

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO



Performance Improvement of a Buck Converter using Kalman Filtering

Adriano Pereira

Mestrado Integrado em Engenharia Eletrotécnica e de Computadores

Supervisor: Prof. Cândido Duarte (FEUP)

Co-Supervisor: Witold Gora (Infineon Technologies AG)

October 30, 2015

A Dissertação intitulada

“Performance Improvement of a Buck Converter using Kalman Filtering”

foi aprovada em provas realizadas em 22-10-2015

o júri



Presidente Professor Doutor Rui Manuel Esteves Araújo
Professor Auxiliar do Departamento de Engenharia Eletrotécnica e de Computadores
da Faculdade de Engenharia da Universidade do Porto



Professor Doutor Rui Manuel Escadas Ramos Martins
Professor Auxiliar do Departamento de Eletrónica, Telecomunicações e Informática
Universidade de Aveiro



Professor Doutor Manuel Cândido Duarte dos Santos
Professor Auxiliar Convidado do Departamento de Engenharia Eletrotécnica e de
Computadores da Faculdade de Engenharia da Universidade do Porto

O autor declara que a presente dissertação (ou relatório de projeto) é da sua exclusiva autoria e foi escrita sem qualquer apoio externo não explicitamente autorizado. Os resultados, ideias, parágrafos, ou outros extratos tomados de ou inspirados em trabalhos de outros autores, e demais referências bibliográficas usadas, são corretamente citados.



Autor - Adriano Jorge Moreira Pereira

Abstract

Switching mode power supplies (SMPS) are widely used in power management systems due to the high efficiency that they can achieve. When comparing the two methods of controlling an SMPS, i.e. voltage-mode and current control, the latter offers an easier compensation loop implementation and faster response to load changes. However, a current sensor is required, at the inductor or the input. This sensing scheme increases the overall bill of materials of the system, the converter footprint, and makes the system more sensitive to several noise sources.

With the constant development of integrated circuits such as microcontrollers, it is possible nowadays to develop new control techniques in terms of hardware and software synergies. This provides a shift in performance to higher levels by the deployment of advanced digital signal processing algorithms, breaking the barriers imposed by technological limits of the today's semiconductor hardware components.

The present dissertation aims at investigating the implementation of an extended Kalman filter (EKF) to improve the performance of a synchronous buck converter. The proposed EKF is based on a mathematical model of the buck converter, which takes into consideration several parasitics such as the inductor and capacitor equivalent series resistors. The goal is to take the advantages of using an observer type filter as the EKF in order to reduce the impact of noise sources, for instance thermal noise or any other type of Gaussian distributed noise. In the proposed EKF approach, the inductor current is estimated with no additional hardware and used as a parameter for the predictive control.

The EKF algorithm has been implemented on an Infineon XMC microcontroller controlling a synchronous buck converter. Experimental results demonstrate an accurate estimation of the inductor current and reveal fast step responses.

Resumo

Fontes de Alimentação de modo de comutação (FAMC) são vastamente usadas em sistemas de gestão de energia devido à grande eficiência que conseguem atingir. Comparando os dois métodos de controlar uma FAMC, i.e. o modo de tensão e o modo de corrente, o último oferece um anel de compensação mais rápido e uma resposta mais rápida quando a carga é alterada. No entanto este método requer um sensor de corrente na entrada ou na bobine. A medição de corrente aumenta o custo de produção do sistema e torna o sistema mais sensível a várias fontes de ruído.

Com o desenvolvimento constante de circuitos integrados como microcontroladores, hoje em dia é possível desenvolver novas técnicas de controlo tirando partido das sinergias entre hardware e software. Estas técnicas proporcionam um aumento do desempenho, pois permitem desenvolver algoritmos avançados de processamento de sinal, quebrando as barreiras impostas pelos limites tecnológicos dos componentes de hardware existentes hoje em dia.

Esta dissertação tem como objetivo investigar a implementação de um filtro de Kalman estendido para melhorar a performance de um buck converter síncrono. O filtro de Kalman estendido proposto é baseado num modelo matemático do conversor buck, que tem em consideração parâmetros parasitas tais como as resistências em série equivalentes da bobine e do condensador. O objetivo é tirar partido das vantagens de usar um observador tal como o filtro de Kalman estendido de modo a reduzir o impacto das fontes de ruído, como por exemplo ruído térmico ou qualquer outro tipo de ruído Gaussiano. Usando a abordagem proposta a corrente na bobine é estimada sem qualquer hardware adicional e usada como parâmetro para controlo preditivo.

O algoritmo do filtro de Kalman estendido foi implementado num microcontrolador Infineon XMC, usado para controlar um conversor buck síncrono. Resultados experimentais demonstram uma estimação exata da corrente na bobine e revelam rápidas respostas ao degrau.

Acknowledgments

First of all I would like to thank my supervisor, Cândido Duarte, for his support, availability and scientific experience and to my co-supervisor, Witold Gora for the industrial background and guidance through all the thesis.

Moreover, I would like to thank all the engineers from the industrial microcontrollers division from Infineon Technologies, for receiving me during part of the thesis development. In special, I would like to thank Eng. Pedro Costa for his guidance and constant support in the diverse phases of the thesis and Eng. Pablo Yélamos for his valuable inputs.

Furthermore, I wish to express my gratitude to all my family and friends that supported and helped me not just during this dissertation, but during all my life.

Finally I would like to thank my girlfriend Mariana for her constant support and for her patience during all this time.

Adriano Pereira

*“Learn from yesterday, live for today, hope for tomorrow.
The important thing is not to stop questioning.”*

Albert Einstein

Contents

Abstract	iii
Resumo	v
1 Introduction	1
1.1 Problem Statement	3
1.2 Proposed Solution	4
1.3 Objectives of this Dissertation	4
1.4 Structure of the Dissertation	5
2 Overview of Buck Converter Control	7
2.1 Voltage Mode Control	7
2.2 Current Mode Control	9
2.3 Predictive Current Control	12
2.4 Summary	14
3 EKF Based Predictive Current Control	17
3.1 Modeling the Buck Converter	17
3.2 Extended Kalman Filter Design	19
3.3 Control Algorithm	23
3.4 Summary	26
4 Experimental Results	27
4.1 Setup Description	27
4.2 Current Estimation Simulations	30
4.3 Implementation of the Algorithm in the Microcontroller	33
5 Conclusions and Future Work	37
5.1 Fulfillment of the Objectives	37
5.2 Future Work	38

List of Figures

1.1	Synchronous buck converter and its waveforms	2
2.1	Block diagram of the buck-converter control	8
2.2	Current mode control	9
2.3	Perturbated inductor current behavior	11
2.4	Boost converter control algorithm	14
3.1	Synchronous buck converter circuit with parasitics	18
3.2	Current waveform considering the proposed method	24
3.3	Proposed predictive current control loop	25
4.1	Infineon XMC buck converter kit	28
4.2	Microcontroller peripheral usage	29
4.3	Simulation of current and resistor estimation with fixed load	31
4.4	Simulation of current and resistor estimation with step load	32
4.5	Simulation of current and resistor estimation with step load and dynamic error covariance update	32
4.6	Output voltage values affected by different magnitudes of noise	33
4.7	Accuracy of EKF estimations in steady state	34
4.8	Voltage and current step response to a step load	35
4.9	Measurement of current and resistor estimation with step load	35
4.10	Voltage response to a sudden input voltage drop	36

List of Tables

1.1	Brief description of the problem	3
1.2	Brief description of the proposed solution	4
2.1	Pros and cons of power converter control modes	11
2.2	Brief comparison of control modes	15
4.1	Buck converter design specifications	27
4.2	Buck converter parameter specifications	27

Abbreviations

ACC	Average Current Control
ADC	Analog-to-Digital Converter
CCM	Continuous Conduction Mode
CCU8	Capture/Compare Unit 8
CMC	Current Mode Control
CS	Control Switch
DCM	Discontinuous Conduction Mode
DSP	Digital Signal Processor
EKF	Extended Kalman Filter
EMI	Electromagnetic Interference
ESR	Equivalent Series Resistor
FPGA	Field-Programmable Gate Array
IDE	Integrated Development Environment
KF	Kalman Filter
HRPWM	High Resolution Pulse Width Modulation
PCB	Printed Circuit Board
PCC	Peak-Current Control
PI	Proportional-Integral
PWM	Pulse Width Modulation
SMPS	Switching Mode Power Supply
SS	Synchronous Switch
SSA	State Space Averaging
VADC	Versatile Analog-to-Digital Converter
VCC	Valley-Current Control
VMC	Voltage Mode Control

Chapter 1

Introduction

One of the most important problems that electrical engineers had to solve was the energy transportation, from production site to the end users, such as factories, public illumination or houses. In order to provide an efficient transportation (with minimum power losses) engineers came to the conclusion that one solution could pass by having different levels of voltages across the distribution network. Nowadays, those levels normally vary between hundreds of kV and a few hundreds of V, with three or four layers in between. This implies the need for devices that are capable of increasing/decreasing the voltage across the distribution network. Those devices are part of an important area in electrical engineering, which is power conversion.

Power conversion, however, is not limited to energy transportation systems, which mainly rely on AC-AC converters (transformers). Power conversion systems can be also AC-DC, DC-AC or DC-DC. These last converters can be categorized in linear regulators and switching-mode power supplies (SMPS). Until recent years, linear regulators were the most common solution used, featured by low ripple and fast transient response. The main problems with linear regulators are their adaptability and the efficiency, which is relatively low. For instance, for a typical input-to-output ratio of 12 V to 3.3 V, the efficiency is in the order of $3.3/12 \simeq 28\%$ [1]. On the other hand, modern SMPS can theoretically achieve an efficiency of 100% at any voltage ratio, meaning that if the target is power efficiency, then SMPS are the answer. Numerous topologies exist for the design of SMPS, such as buck, boost, buck-boost, half-bridge or full-bridge. The most known and the one studied in this dissertation is the buck converter.

The most classic circuit for a buck converter comprises four main components: an inductor, a capacitor, a transistor acting as a control switch (CS) and a freewheel diode, which conducts when the switch is off. To further improve the efficiency, the freewheel diode is replaced by another switch, with the purpose of reducing the power losses of the diode with a lower conduction resistance of the additional switch [2]. This switch, usually named synchronous switch (SS), only conducts when the main switch is not conducting, requiring a synchronization between both control signals. The diagram of a *synchronous* buck converter is depicted in the Fig. 1.1a. For the present dissertation, only the operation in continuous conduction mode (CCM) will be assumed, i.e. when the buck converter operates with the inductor current always above zero, which

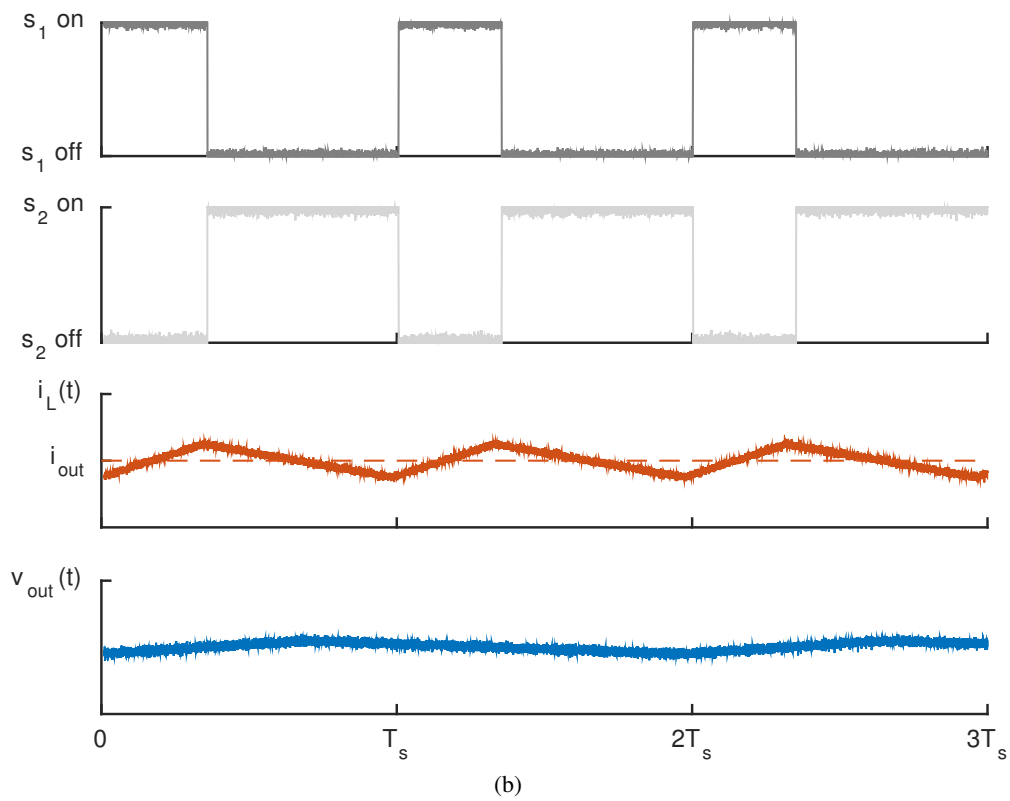
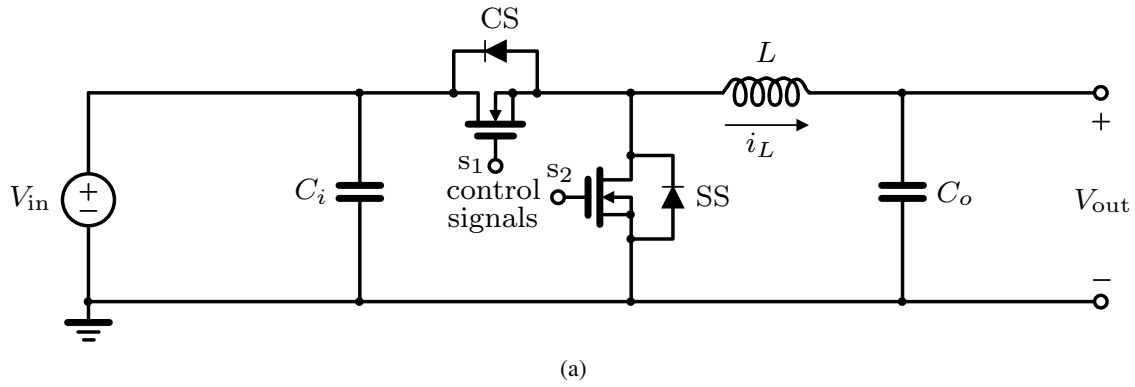


Figure 1.1: Synchronous buck converter (a) circuit, and (b) illustrative waveforms.

is its most common usage – discontinuous conduction mode (DCM) will not be addressed here. The inductor current and output voltage relation for several switching times T_s are represented in Fig. 1.1b.

The elimination of the noisy contributions in SMPS can be quite troublesome. In fact, noise can manifest due to several sources in a buck converter. Input noise can be dominant because it can be propagated directly into the output. This can be due to the finite input capacitance or a significant value of its equivalent series resistance (ESR) [3]. Other noise mechanisms originate from a non-optimized design of the printed circuit board (PCB). For instance, input-to-output noise injection can arise from the input lead inductance, which in conjunction with the input capacitor

Table 1.1: Brief description of the problem.

	Problem	
	+	–
Current-mode control		Faster transient response
		Requires current sensor, which increases cost

can cause ringing effects. At the output, also the finite output capacitance can represent a noise source, as well as its ESR, which lead to an undesirable ripple behavior [3]. In general, the suppression of such noise contributions is not straightforward. The probability distribution of the noise processes mentioned above is not usually known. Some noise can be Gaussian if associated with conductor resistivity, but most of the phenomena mentioned mainly produce a constant ripple behavior at fixed frequency. Typically, the latter case is often mitigated by using good quality (expensive) passive devices in conjunction with a careful layout of the PCB. On the other hand, eliminating Gaussian noise is not easy, since it is not concentrated at a single frequency. This problem is also addressed in the present dissertation.

1.1 Problem Statement

The most intuitive way for controlling the output voltage is to measure the output voltage, compare it to the predefined reference voltage, and introduce a negative feedback loop into the system – this is generally called voltage-mode control. Another way to maintain a constant voltage at the output consists on controlling the inductor current value (peak, valley, average, etc.) [4]. Electrical principles tell us that current is faster than its voltage, so if the current changes are sensed first than voltage, then it should act on the control loop much faster. For these reasons, in industry current-mode control is often preferred to improve the transient response, and reduce the controller complexity as well.

On a buck converter, the current is typically sensed on the inductor. In order to measure a current in an electronic circuit, a shunt resistor and a high-gain amplifier is typically employed, which due to the current discontinuities must be extremely wide band to avoid signal distortion. Alternatively, a current transformer can also be used, but in any case, the overall production cost is increased. Furthermore, the existence of current sensing circuitry also increases the system footprint and power dissipation. The present work specifically addresses these limitations by using a “current sensorless” approach in which the hardware for current sensing can be omitted and current can still be estimated. As the sensorless current estimation avoids additional hardware, noise immunity is required in the estimation process, otherwise the resulting current will be inaccurate. Table 1.1 provides a summary for the problem addressed.

Table 1.2: Brief description of the proposed solution.

Proposed solution	
Approach:	Predictive current control based on current estimation with EKF
Targets:	Better performance than conventional voltage-mode control with the same number of components
	Improved noise immunity inherent to KF processing, with similar performance compared to other predictive methods, and similar computation requirements as well.

1.2 Proposed Solution

In recent years some methods were proposed to overcome the aforementioned drawbacks. To eliminate the need for the current sensing hardware, prediction techniques have been developed for the estimation of parameters of the buck converter. For the prediction of the inductor current, current observers can be used [5]. However, for typically noisy systems such as non-isolated converters, estimation based on Kalman filtering (KF) is preferable [6].

KF is a state estimator, introduced in 1960 by Rudolf Kalman [7], which allows the extraction of useful information in noisy signals. It can be regarded as a kind of observer, although it relies on a stochastic process rather than deterministic. For nonlinear processes, extended Kalman filtering (EKF) is usually adopted. This dissertation proposes a method to implement a current control algorithm in a synchronous buck converter using EKF. In the proposed solution the inductor current is estimated, avoiding the use of current sensor hardware. Moreover, since the proposed approach relies on KF, as long as the noise is Gaussian distributed, noise reduction can also be achieved when compared with other types of predictive approaches. Table 1.2 gives a brief summary of the proposed solution.

1.3 Objectives of this Dissertation

SMPS were originally controlled with analog loops. However, with increasing processing capability at reduced areas and more competitive prices of integrated circuits such as microcontrollers, digital signal processors or field-programmable gate arrays, led to an increasing growth in the digital control methods. These allow the use of more complex functions, which cannot be implemented in fully-analog approaches [8]. The present dissertation addresses current-mode control performed in the digital domain with a microcontroller from Infineon Technologies. In specific, the objectives for this dissertation can be summarized as follows:

- Investigate and develop a current estimation algorithm, based on a model of the buck converter circuit and its input and output voltages;
- Design and tune a control loop that uses the current estimation algorithm to control a synchronous buck converter;

- Implement an optimized and efficient control loop in a experimental setup, in order to prove the theoretical concept.

The proof of concept is implemented in the a buck converter evaluation kit from Infineon Technologies, which uses the XMC4200 microcontroller (ARM-M4-based architecture). To aid in the code development and debug, the new DAVE4 IDE is used, launched as productive version in July 2015 by Infineon.

1.4 Structure of the Dissertation

Following the brief introduction to the dissertation goals, the present document is organized as follows.

- Next chapter contains an overview of SMPS control methods an their evolution. Voltage control mode, current control mode and its sensorless implementation, the predictive current control are described, analyzed and compared. The literature review for predictive current control is also done in this chapter.
- Chapter 3 provides a theoretical derivation and development of a predictive current control algorithm based on EKF. The model of a synchronous buck converter, the EKF tuning and the control loop design will be described.
- In Chapter 4, the simulations and proof-of-concept experiments are presented, including a description of the hardware and conditions of operation of the converter. A comparison is performed between the expected values (from the analysis in Chapter 3) and the measurement results.
- The final chapter gives a critical analysis of the obtained results, as well as a comparison with other methods present in the literature. Finally, the fulfillment of objectives is discussed, followed by suggestions for future work.

Chapter 2

Overview of Buck Converter Control

Control methods for SMPS evolved through the years in order to adapt to market needs. Most classic approaches fall into two main categories, which are the voltage-mode control (VMC), where the duty cycle is proportional to the control voltage, and current-mode control (CMC), where the converter operates as a voltage-controlled current source. The advantages, disadvantages and evolution of each of the mentioned modes of control will be analyzed in the first two sections. The third section introduces the predictive current control, which tries to overcome some problems that occur on CMC. Finally in the last section a summary of the chapter is done, presenting in a compact way the methodologies, as well the pros and cons of the presented methods.

2.1 Voltage Mode Control

VMC has been perhaps the most common method applied not only to buck converters, but also to numerous SMPS topologies with less stringent requirements. The reason for this is merely the inherent simplicity of design and implementation [9, 10]. The working principle of the control loop consists on comparing error between the output voltage and the reference voltage to a constant ramp waveform with fixed frequency, whereas the resultant error signal controls the duty cycle by means of PWM¹. Therefore, VMC just needs a single feedback loop to effectively regulate the output voltage.

Measuring the output voltage, however, means that if any change on the load occurs such change needs to be properly sensed and then corrected by the control loop. But sensing the load variation is not an easy task as it may seem. This is because such power system acts as a low impedance output (or as voltage source), hence a load change does not have a immediate impact on the output voltage, slowing down the transient response. Referring to Fig. 2.1, the control

¹PWM generation based on ramp waveform comparison might be seen as a strictly analog approach. However, equivalent ways to implement the duty-cycle control can be done also in the digital domain, although such alternative will not be detailed here because the operating principle remains practically the same.

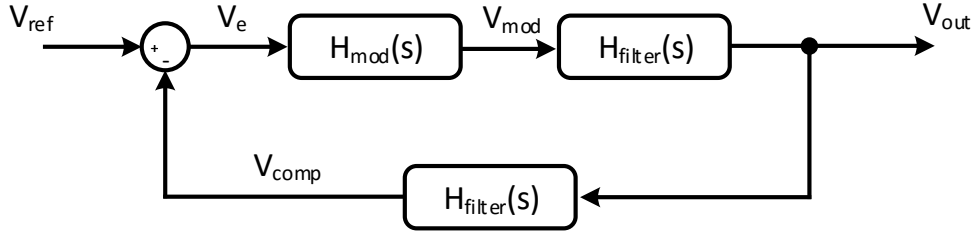


Figure 2.1: Block diagram of the buck-converter control.

transfer function of a VMC buck converter is given by

$$\frac{V_{out}(s)}{V_{ref}(s)} = \frac{H_{mod}(s) \cdot H_{filter}(s)}{1 + H_{loop}(s)} \quad (2.1)$$

where the loop response can be related as follows [11]

$$\frac{H_{loop}(s)}{H_{comp}(s)} = H_{mod}(s) \cdot H_{filter}(s) = \frac{V_{in}}{V_{ramp}} \cdot \frac{1 + s/\omega_z}{1 + (s/\omega_0)/Q + (s/\omega_0)^2} \quad (2.2)$$

where the modulator voltage gain is given by $H_{mod}(s) = V_{in}/V_{ramp}$ [12]. The remaining parameters in (2.2) refer to the output filter transfer function. Here, for illustrative purposes, an ideal passive network was assumed, except for the capacitor that presents an equivalent series resistance R_{ESR} . The resonance frequency is defined by $\omega_0^2 = LC$, the non-null value of R_{ESR} is responsible for a zero in the transfer function $s = -\omega_z = -1/(R_{ESR}C)$, and the quality factor of the parallel RLC at the output is defined by $Q = R_{out}/(\omega_0 L)$, with R_{out} representing the load. Note that the LC introduces two conjugate poles, causing a 180 degree phase lag at ω_0 . As the loop function transfer is given by $H_{loop}(s) = H_{mod}(s) \cdot H_{filter}(s) \cdot H_{comp}(s)$ [13], such abrupt phase shift is included also in the loop transfer function. Hence a low-frequency dominant pole is usually required to compensate such effects that could cause ringing and instability [11]. This, as a consequence, also limits the speed of the converter. A typical response time in VMC buck converters may be in the order of tens of switching cycles [9]. It should be also mentioned an additional difficulty in compensation, which is the fact that the loop gain is affected also by the input voltage itself [10].

Besides the speed limitations and compensation challenges, VMC also poses other obstacles. In modern buck converters, it is usual to find some protection circuit against high currents for reliability purposes. Due to the fact that VMC only takes into account the input and output voltages, the implementation of an overcurrent protection circuit is not straightforward [10]. In summary, in spite of its simplicity, there are several drawbacks in VMC that are not easily tolerable in modern converter systems. This motivates the use of alternative control methods such as CMC, which is addressed next.

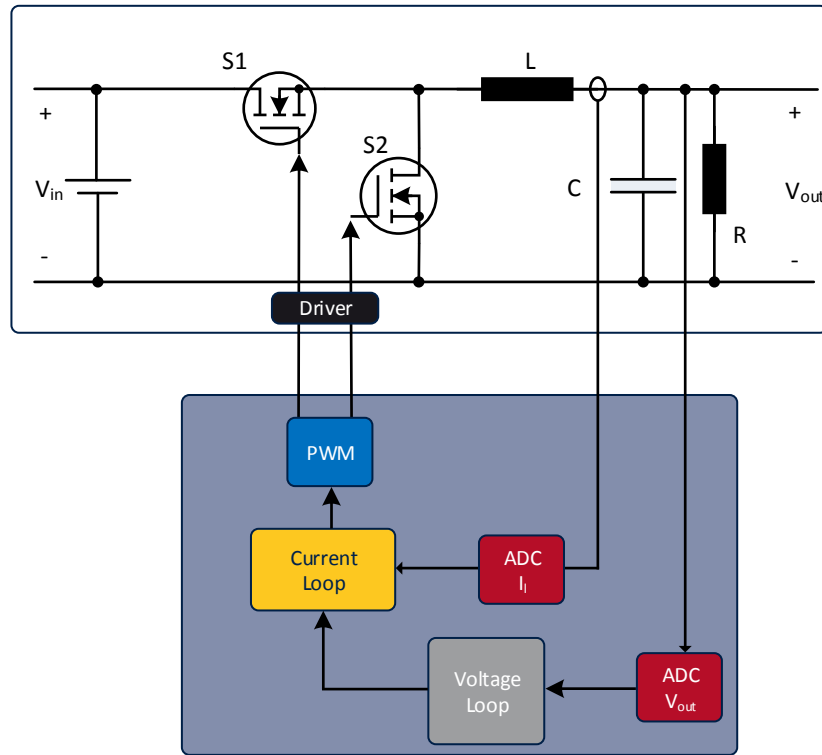


Figure 2.2: Current mode control.

2.2 Current Mode Control

The problems just mentioned that occur in VMC can be somehow circumvented using CMC. In order to achieve the desired output voltage, as the name suggests, this method controls the current – the inductor current $i_L(t)$. In a buck converter, the inductor current is proportional to $V_{in} - V_{out}$, meaning that any change on the input or output voltages will be immediately sensed, improving the control loop speed [10]. This is done, as depicted in Fig. 2.2, using two control loops: the voltage control loop (typically denoted as “outer loop”), which monitors the converter output voltage, and the current control loop (“inner loop”) that controls the current at the inductor by changing the switch state when some reference value is reached.

In addition to a faster response, CMC has also cycle-by-cycle current limiting, which overcomes some issues previously mentioned with VMC. However, one obvious drawback of this method is the need for an additional control loop, which may sound a more complex process but, in fact, the compensation obstacles are more relaxed than in the VCM approach. For instance, since the inductor behaves as a current source, the double pole turns into real poles dictated by the output capacitor and load resistance [12].

For the current sensing, however, additional circuitry is demanded, which translates into additional production costs as well [9, 10]. For convenience, current sensing in the buck converter is sometimes performed at the control switch rather than at the inductor [14]. In other implementations the current sensing is performed using a shunt resistor with a small ohmic value placed in series with the inductor. The voltage drop across the resistor is applied at the inputs of a high-gain amplifier and forwarded to the analog-to-digital converter of the microcontroller. Other alternative consists in using a current transformer or a hall-effect sensor, which reduces the power losses, at an increased manufacturing cost.

Typically, the switching frequency of power supplies is in the range of hundreds of kHz and recently there are many reaching the range of MHz. Such high frequency switching causes glitches in the inductor current waveform. Different CMC methods were developed to overcome this issue, namely:

- Peak-current control (PCC);
- Valley-current control (VCC);
- Average-current control (ACC).

PCC senses the peak inductor current and compares it to a reference level generated by the outer loop. The moment when the inductor current hits the current reference determines the moment that the control switch is turned off. One problem of PCC occurs at the time instant that the switch turns on. At this moment, parasitic capacitances present in the switching components cause a spike in the inductor current, which might hit the desired current prematurely, turning off the switch too soon. One common solution is performing the leading edge blanking, avoiding current sampling during the time that the glitch may occur, resulting a minimum dead time for the SMPS regulation [15].

Furthermore, at duty cycles above 50 %, this method becomes unstable and results in sub-harmonic oscillation. Sub-harmonic oscillation can occur when the inductor current suffers a perturbation. Considering that the perturbation occurs at the beginning of a cycle, the convergence of the current depends on the value of the PWM duty cycle. Fig. 2.3 depicts the behavior of the steady-state waveform and compares it to the perturbed waveform response.

At duty cycles above 50 %, as can be observed, the current will not converge to the steady state value, causing the oscillations. To damp such oscillation, slope compensation is usually employed, but such approach may result in some performance degradation in response to fast load changes [16].

VCC acts in the opposite way to PCC: the current is sensed when the control switch is off (on the down slope of the inductor current) and controls the turning on of the control switch. Contrarily to PCC, VCC can operate with small duty-cycle values and do not require leading edge blanking. As described by Harriman [17], it also suffers from sub-harmonic oscillation at duty cycles lower than 50 %, requiring slope compensation, which is more difficult to achieve than in PCC. Another disadvantage of VCC is its poor line regulation, compared to PCC, due to the

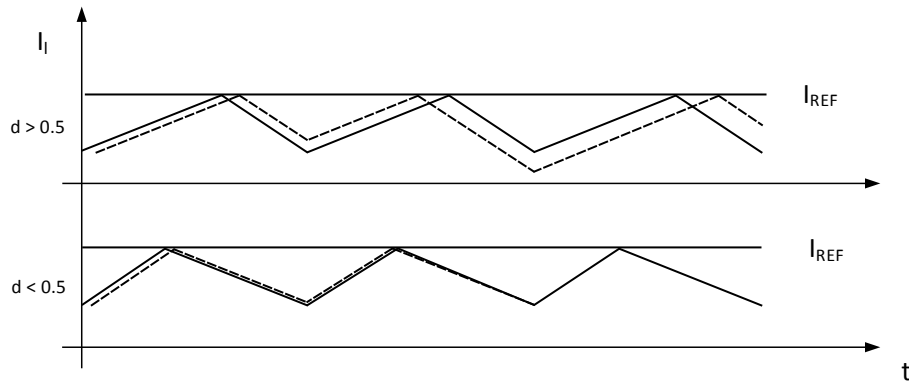


Figure 2.3: Comparison of the steady-state current (full line) with the perturbed current waveform (dashed line) at different duty cycles.

slope that it considers. In PCC, the inductor up slope is considered, being this slope given by $(V_{in} - V_{out})/L$, performing automatically the input voltage feedforward. In VCC, however, the down slope $(-V_{out}/L)$ is considered, which does not contain the input voltage term, degrading then the response to input voltage changes.

ACC overcomes the sub-harmonic oscillation that occurs in both PCC and VCC, so slope compensation is not required. Conversely, the inductor current average value cannot be directly measured without additional circuitry or post processing [4]. Also ACC may not provide cycle by cycle limitation, contrarily to PCC and VCC.

Table 2.1 summarizes the pros and cons of the referenced control methods.

Table 2.1: Pros and cons of power converter control modes.

Method	Pros	Cons
VMC	Simplicity No additional components required	Slow response to load variations Difficult to compensate Poor line regulation
PCC	Fast transient response Cycle-by-cycle current limitation Good line regulation	Oscillations at $d > 50\%$ Needs slope compensation Needs a comparator blanking mechanism Has a minimum duty cycle
VCC	Cycle-by-cycle current limitation No need for blanking of the comparators	Oscillations at $d < 50\%$ Needs slope compensation Poor line regulation Slower than PCC
ACC	No sub-harmonic oscillations	Slow transient response Difficult to measure the average value

2.3 Predictive Current Control

The classical control methods described in the last two sections are the most used and the most mature today. In the recent years, a new trend appeared, which instead of measuring the inductor current, tries to estimate it, eliminating the need for additional sensors – this is called predictive current control.

Predictive current control uses the often called current observers to substitute the current measurement circuit. A current observer is nothing more than an algorithm that uses a model of the system and measurements of one or more system variables to estimate the current value. In the case of the buck converter, normally the input and output voltages are measured and used to estimate the inductor current value.

One of the first methods that tried to predict the inductor current was presented in [18], which relied on the measurement of the inductor voltage. This voltage would then be integrated and represent the inductor current value. Nowadays, two main methodologies are employed to derive a model for the inductor current: one is based in the known equations for the inductor current slopes, which are calculated based on buck converter measured voltages (input and output) and circuit intrinsic parameters, such as the inductor value; the other one, instead of modeling only the current, takes into account the model for the complete system. Usually the second approach is computationally expensive when compared to the first one, because more variables are considered in addition to the one that needs to be estimated. Buck converters, as any other electronic circuit, has its own non-idealities and parasitics due to the components and intrinsic work principle, being the modeling of the complete system an advantage – the cross influences are better represented.

Next, a review of existent methods is presented for predictive current control, using both methodologies described above. The performance of these methods will be evaluated and compared, not only between them, but also with state of the art control methods (VMC and CMC).

A digital control method for DC-DC boost converters operating in CCM is studied by Mattavelli [19]. The resistive parasitics of the circuit are not considered, so the derived model for the system only estimates changes in the value of the inductor current. Prediction of the inductor current variations is done through a state observer, measuring in the first implementation only the output voltage. The proof-of-concept algorithm was implemented in a FPGA and developed in VHDL, being the hardware a 100 W boost converter, with an input voltage of 24 V, output of 48 V and switching frequency of 180 kHz. Experimental results show good adaptability of the method to load variations but struggles with input voltage disturbances, which lead to the implementation of an input voltage feed-forward algorithm. Simulations and experimental results proved the better performance over the initial method to input voltage changes. The results were also compared with a conventional voltage mode control method, showing better performance when the input voltage is feed-forwarded.

Two methods that do not require current measurement are proposed by Beccuti *et al.* [20]. The method extracts the current information using the known system variables, such as the input voltage, output voltage and the duty cycle. In the second approach the authors propose an EKF to

estimate the inductor current and the capacitor voltage and claim that this compensates the computation time delay, accounts the switching behavior of the converter and reduces the measurement noise. The experimental algorithms are implemented in a fixed-point DSP that control a boost converter operating at 100 kHz and with variable input and output voltages. Both algorithms show good adaptability to line and load changes, while the algorithm implemented with EKF is capable to provide a more accurate and less noisy current estimation.

Qiu *et al.* [21] propose a general approach to 3 different types of SMPS – buck, boost and buck-boost converters – operating in continuous conduction mode. The average current control strategy tries to implement a first-order low pass filter based on the sampling of the input and output voltages to estimate the current, using the last one to compare with the current reference provided by the voltage loop. Current regulation is achieved in two steps: in the first switching cycle the valley current is adjusted, being the average value regulated in the subsequent cycle. Experimental results are evaluated in a boost converter, being the input and output voltages, respectively, 12 V and 24 V, with a 100 kHz switching frequency. A TMS320F2812 DSP system was used as control hardware. The results showed a 200 μ s step response to both step load or input voltage changes. In addition to this fact, the current reference is achieved in two switching cycles with the proposed control algorithm.

A similar method is presented in [5], proposing a current observer for predictive current control of a buck converter. In the first implementation the current observer is based in the equations of the inductor current slopes, without considering any parasitics in the system. The author observes that with the ideal equations the method would diverge and proposes an optimal current observer, having into consideration the parasitic parameters of the circuit, such as the freewheel diode voltage drop and the inductor series resistance, achieving the desired convergence of the method. The control loop used in [5] applies controls the valley current with trailing edge modulation, claiming that it can eliminate the current error in less than two switching cycles. The experimental setup consists in a buck converter, controlled by a DSP that steps down the voltage from 10 V to 6 V with a switching frequency of 100 kHz. Results indicate that the step response of the proposed method rounds the 200 μ s, showing a 2 times faster response when compared to a regular VMC loop, implemented using a PI controller.

Zhang *et al.* [22] develop a method using an EKF for estimating the inductor current in a boost converter. The boost converter is modeled using state-space averaging and in the first implementation modeled with a fixed load. This modeling is not suitable to a buck converter in normal mode of operation, because the load is always changing (otherwise there would not be a necessity for the control loop). To overcome this issue, a load estimation algorithm is proposed, rewriting the load as a function of the inductor current and the output voltage. The complete control algorithm is depicted in Fig. 2.4.

The proof-of-concept is done in a boost converter that steps up the current from 5 V to 15 V with a switching frequency of 100 kHz. Experimental results prove that the method is able to estimate the current with a 5 % error within steady state and 10 % error in the transient response of 33 %. Finally the author highlights the fact that this method is less affected by electromagnetic

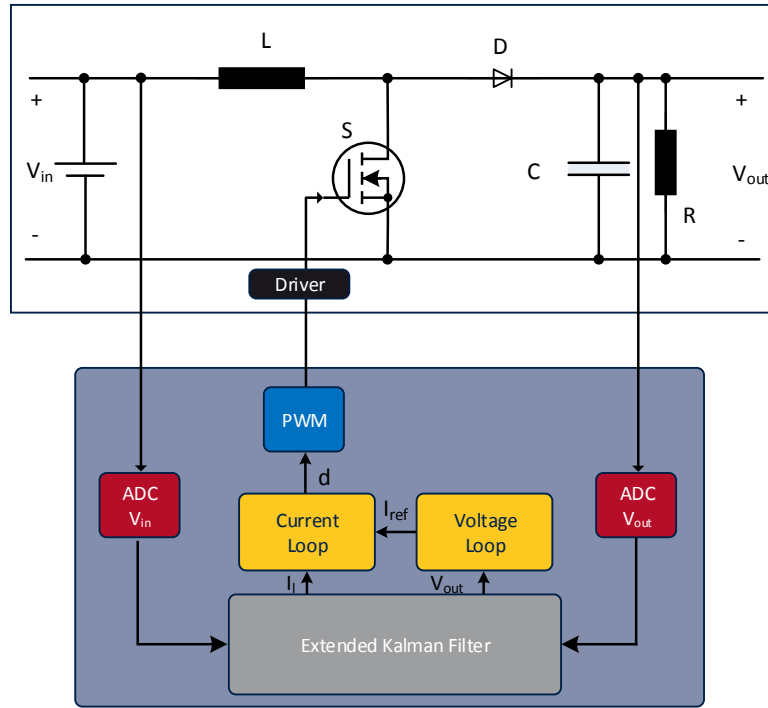


Figure 2.4: Boost converter control algorithm. [22]

interference (EMI) noise than other predictive control methods, giving the fact that EKF can effectively filter out the noise from the output voltage, reducing then its influence on the control loop. This premise, however, is not verified using any simulation or experimental result.

2.4 Summary

An overview of the buck converter control methods was performed in this chapter. In sections 2.1 and 2.2 the classic control modes for SMPS were described. VMC is the most common one due to its simplicity (only has one control loop) but shows slow transient response and doesn't provide a natural current limitation. CMC overcomes the voltage mode limitations sensing the inductor current, which increases the overall cost of production of the device. Table 2.2 summarizes the principles of operation of the different methods.

Different implementations of CMC also have limitations: PCC and VCC suffer from sub-harmonic oscillation at some duty cycles, requiring slope compensation and have an inherent limitations on duty cycles. Average-Current Control solves some of the problems of PCC and VCC but the measurement of the average current is more difficult.

The drawbacks in the current measurement led to a different digital method for controlling SMPS: the Predictive Current Control. Mattavelli [19] proposes a model for the variations of

Table 2.2: Brief comparison of control modes.

Method	Operation principle
VMC	Duty cycle proportional to output voltage
CMC	Converter acts as a current source
• PCC	Control switch turned off when peak i_L is achieved
• VCC	Control switch turned on when valley values i_L is achieved
• ACC	Average current value of i_L tries to achieve the reference value

the system and uses it to control a boost converter. Experimental results show that the control algorithm achieves better results than the voltage mode control only if the input voltage is feed-forwarded. Beccuti *et al.* [20] present two different methods of controlling DC-DC converters: in the first the inductor current is derived from the system known or measured variables, such as the duty cycle and the input voltage and output voltage; the second one considers the whole system model and is implemented using EKF. This approach shows better accuracy and noise immunity in the current estimation. Qiu *et al.* [21] propose a generalized method for buck, boost and buck-boost topologies, verifying the algorithm with a boost converter. Results show a step response of 200 μ s and a current error elimination in two switching cycles. Run *et al.* [5] present a similar method that uses the inductor current slopes equations to control a buck converter. It achieves a transient response two times faster than the regular voltage mode controller using a PI. Zhang *et al.* [22] use an EKF as a current observer for a boost converter, proposing a load estimation algorithm to improve the results. An accuracy of 5 % is achieved in steady state while in steady state the load error is never bigger than 10 %.

Chapter 3

EKF Based Predictive Current Control

Predictive current control is a method to control SMPS, but unlike other control methods it is not limited only to the design and tuning of control loops. This chapter provides the theoretical derivation of a control loop for a synchronous buck converter in three parts. In the first section a model of the buck converter is presented, considering some important parasitic parameters that try to represent the circuit with reasonable accuracy. On the second section an augmented state EKF for both inductor current and load estimation is proposed. Furthermore, the equations for the state propagation and output are derived. The third section studies the control loop design and tuning using the current observer developed in the first two sections.

3.1 Modeling the Buck Converter

The typical operation of SMPS is considered as “time variant” due to the fact that a different time-invariant system can be derived for each switching cycle [23]. One way to simplify the analysis and modeling of SMPS is applying state-space averaging (SSA). This method has been around since the seventies [24, 25] and since then it has found a relatively widespread use. Besides our target topology, which is a buck converter operating in CCM, SSA has been applied to almost any sort of switching-mode power converter [23, 26].

The approach relies on the assumption that the circuit time constants are much longer when compared to the switching frequency $1/T_s$. From this, a simple but relatively accurate model can be obtained for DC and small-signal analyzes of buck converters. The technique can be applied as follows. First, the system is formulated as in any other linear system, i.e. one defines the states of the system, which usually are voltages or currents of the elements that can store energy, such as capacitors and inductors, respectively – e.g. $\mathbf{x}(t) = [i_L(t), v_C(t)]^T$. Considering the synchronous buck converter represented in Fig. 3.1, during t_{on} the Kirchhoff current and voltage equations are as follows.

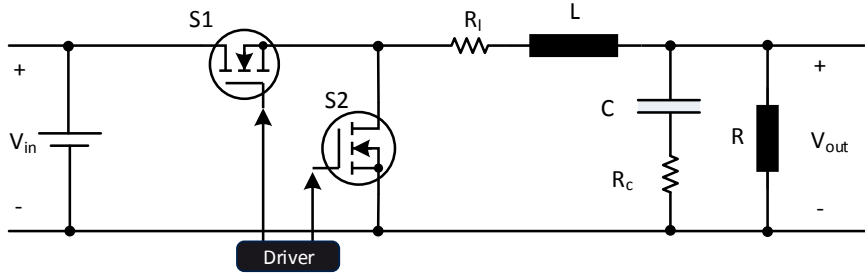


Figure 3.1: Synchronous buck converter circuit with parasitics (R_l and R_c).

$$v_{in}(t) = R_L v_L(t) + L \frac{d}{dt} i_L(t) + v_C(t) + R_C C \frac{d}{dt} v_C(t) \quad (3.1)$$

$$i_L(t) = C \frac{d}{dt} v_C(t) + \frac{1}{R_{out}} \left(v_C(t) + R_C C \frac{d}{dt} v_C(t) \right). \quad (3.2)$$

This set of differential equations express the evolution of the capacitor voltage $v_C(t)$ and the inductor current $i_L(t)$ across the time. Representing the equations (3.1) and (3.2) in order to $\frac{d}{dt} i_L(t)$ and $\frac{d}{dt} v_C(t)$, respectively, the result is

$$\frac{d}{dt} i_L(t) = - \left(\frac{R_L}{L} + \frac{R_C R_{out}}{L(R_{out} + R_C)} \right) i_L(t) - \frac{R_{out}}{L(R_{out} + R_C)} v_C(t) + \frac{v_{in}(t)}{L} \quad (3.3)$$

$$\frac{d}{dt} v_C(t) = \frac{R_{out}}{C(R_{out} + R_C)} i_L(t) - \frac{1}{C(R_{out} + R_C)} v_C(t). \quad (3.4)$$

KF relies on an algorithm that uses the state-space form to perform the estimation of the states. Equations (3.3) and (3.4) are formulated in a way that it is very simple to identify the state-space variables, i.e. the states $\mathbf{x}(t)$ are $i_L(t)$ and $v_C(t)$, and the input to the system $\mathbf{u}(t)$ is $v_{in}(t)$. Writing the mentioned equations in the state-space form, using as reference the general state-space equation $\dot{\mathbf{x}}_1(t) = \mathbf{A}_1 \mathbf{x}(t) + \mathbf{B}_1 \mathbf{u}(t)$, the state space model during t_{on} is given by

$$\begin{bmatrix} \frac{d}{dt} i_L(t) \\ \frac{d}{dt} v_C(t) \end{bmatrix} = \begin{bmatrix} - \left(\frac{R_L}{L} + \frac{R_C R_{out}}{L(R_{out} + R_C)} \right) & - \frac{R_{out}}{L(R_{out} + R_C)} \\ \frac{R_{out}}{C(R_{out} + R_C)} & - \frac{1}{C(R_{out} + R_C)} \end{bmatrix} \begin{bmatrix} i_L(t) \\ v_C(t) \end{bmatrix} + \begin{bmatrix} \frac{1}{L} \\ 0 \end{bmatrix} v_{in}(t) \quad (3.5)$$

The same analysis is then performed for the circuit during t_{off} . The model is very similar to the one during t_{on} , except for the fact that the signal $v_{in}(t)$ is not considered as an input anymore. If we consider the state-space model during t_{off} to be $\dot{\mathbf{x}}_2(t) = \mathbf{A}_2 \mathbf{x}(t) + \mathbf{B}_2 \mathbf{u}(t)$, then $\mathbf{A}_2 = \mathbf{A}_1$ and $\mathbf{B}_2 = [0, 0]^T$.

Unifying both state-spaces is the next step, using SSA. State variations of the buck converter across one switching period T_s can be represented joining in the same equation the t_{on} and t_{off} state

space representations. Considering $\delta = t_{on}/T_s$ and the equation that describes the whole switching period as

$$\dot{\mathbf{x}}(t) = \delta \dot{\mathbf{x}}_1(t) + (1 - \delta) \dot{\mathbf{x}}_2(t) \quad (3.6)$$

$$= [\delta \mathbf{A}_1 + (1 - \delta) \mathbf{A}_2] \mathbf{x}(t) + [\delta \mathbf{B}_1 + (1 - \delta) \mathbf{B}_2] \mathbf{u}(t), \quad (3.7)$$

The state-space averaged model for a synchronous buck converter considering the parasitic parameters is given by

$$\begin{bmatrix} \frac{d}{dt} i_L(t) \\ \frac{d}{dt} v_C(t) \end{bmatrix} = \begin{bmatrix} -\left(\frac{R_L}{L} + \frac{R_C R_{out}}{L(R_{out} + R_C)}\right) & -\frac{R_{out}}{L(R_{out} + R_C)} \\ \frac{R_{out}}{C(R_{out} + R_C)} & -\frac{1}{C(R_{out} + R_C)} \end{bmatrix} \begin{bmatrix} i_L(t) \\ v_C(t) \end{bmatrix} + \begin{bmatrix} \frac{\delta}{L} \\ 0 \end{bmatrix} v_{in}(t) \quad (3.8)$$

However, the derived model is a continuous time state-space model, which is inappropriate to implement in a discrete control algorithm, such as the one implemented in a microcontroller. A discrete state-space model of $\dot{\mathbf{x}}(t) = \mathbf{A} \mathbf{x}(t) + \mathbf{B} \mathbf{u}(t)$ can be calculated by

$$\mathbf{x}(k+1) = e^{\mathbf{A}T} + \left(\int_{\tau=0}^T e^{\mathbf{A}\tau} d\tau \right) \mathbf{B} \quad (3.9)$$

or by a small-step approximation considering that $e^{\mathbf{A}T} \approx \mathbf{I} + \mathbf{A}T$. Using the approximation, the discrete state-space model of the buck converter is given by

$$\begin{bmatrix} i_L(k+1) \\ v_C(k+1) \end{bmatrix} = \begin{bmatrix} 1 - \left(\frac{R_L}{L} + \frac{R_C R_{out}}{L(R_{out} + R_C)}\right) T_s & -\frac{R_{out} T_s}{L(R_{out} + R_C)} \\ \frac{R_{out} T_s}{C(R_{out} + R_C)} & 1 - \frac{T_s}{C(R_{out} + R_C)} \end{bmatrix} \begin{bmatrix} i_L(k) \\ v_C(k) \end{bmatrix} + \begin{bmatrix} \frac{\delta T_s}{L} \\ 0 \end{bmatrix} v_{in}(k). \quad (3.10)$$

3.2 Extended Kalman Filter Design

KF is applied to linear systems, described in the form of a state space model, generically represented as

$$\mathbf{x}_{k+1} = \mathbf{A} \mathbf{x}_k + \mathbf{B} u_k + \mathbf{w}_k \quad (3.11)$$

$$\mathbf{z}_k = \mathbf{H} \mathbf{x}_k + \mathbf{v}_k \quad (3.12)$$

where \mathbf{x} is the system state variable, \mathbf{u} is the system input vector, \mathbf{z} is the measurement, \mathbf{A} is the state transition matrix, \mathbf{H} is the observation matrix and \mathbf{w} and \mathbf{v} the process and measurement noises, respectively. The state transition matrix \mathbf{A} can be seen as the description of the system dynamics – the relation between the current and the next system state. \mathbf{H} shows the relationship between the measured values and the state of the system.

Given this state model, KF makes use of a state prediction relying on the past estimated states. The algorithm is divided in two main phases, which are the prediction and estimation. Considering the system state \mathbf{x}_k at the instant k , prediction phase takes the information until $k - 1$ and predicts

the state \mathbf{x}_k . The estimation phase accounts the measurement of the state, \mathbf{z}_k and corrects the predicted value.

The prediction phase has into account the system state vector and produces an estimation for both the state and covariance, using

$$\hat{\mathbf{x}}_k^- = \mathbf{A} \hat{\mathbf{x}}_{k-1} + \mathbf{B} \mathbf{u}_k \quad (3.13)$$

$$\mathbf{P}_k^- = \mathbf{A} \mathbf{P}_{k-1} \mathbf{A}^T + \mathbf{Q} \quad (3.14)$$

where \mathbf{P} is the error covariance matrix and \mathbf{Q} is the process noise covariance. The superscript '-' means that the value is predicted, while the '^' superscript means an estimated value.

In the estimation process, three main steps are executed. During the first one, the Kalman gain is calculated as follows

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}^T (\mathbf{H} \mathbf{P}_k^- \mathbf{H}^T + \mathbf{R})^{-1} \quad (3.15)$$

where \mathbf{K} is the Kalman gain and \mathbf{R} is the measurement noise covariance.

The second step computes the state estimation, using the measured value to update the previously predicted state

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k (\mathbf{z}_k - \mathbf{H} \hat{\mathbf{x}}_k^-) \quad (3.16)$$

where \mathbf{z} is the measurement. In this process, the final value of the estimation is calculated.

In the next step, the error covariance is computed. This value needs to be updated for the next cycle iteration

$$\mathbf{P}_k = \mathbf{P}_k^- - \mathbf{K}_k \mathbf{H} \mathbf{P}_k^- \quad (3.17)$$

This concludes the algorithm. A more compact form in algorithm is presented in (3.1).

Algorithm 3.1: Kalman Filter

Init: Intialize $\hat{\mathbf{x}}_0, \mathbf{P}_0$

Input: Measurement \mathbf{z}_k

Output: Estimation $\hat{\mathbf{x}}_k$

1 **for** $k \leftarrow 0$ **to** n **do**

2 $\hat{\mathbf{x}}_k^- = \mathbf{A} \hat{\mathbf{x}}_{k-1} + \mathbf{B} \mathbf{u}_k$

3 $\mathbf{P}_k^- = \mathbf{A} \mathbf{P}_{k-1} \mathbf{A}^T + \mathbf{Q}$

4 $\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}^T (\mathbf{H} \mathbf{P}_k^- \mathbf{H}^T + \mathbf{R})^{-1}$

5 $\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k (\mathbf{z}_k - \mathbf{H} \hat{\mathbf{x}}_k^-)$

6 $\mathbf{P}_k = \mathbf{P}_k^- - \mathbf{K}_k \mathbf{H} \mathbf{P}_k^-$

7 **end**

The model derived in the previous section is the base model for the current estimation algorithm. Nevertheless the dependency of the model on the load resistor R_{out} arises a limitation to this model – in a buck converter the load is not known and varies with time. To overcome this limitation, a estimation of two variables is performed, namely the inductor current i_L and the load resistor R_{out} . One way to do this double estimation is considering R_{out} as an additional state that

will be estimated alongside i_L . The model of the buck converter until now was linear, hence suitable to implement with the standard KF algorithm. But if the load is estimated it will become nonlinear, due to the state multiplication.

EKF becomes then the most appropriate state estimator to apply to this model, considering its capability of handling with nonlinear models. This method assumes a new model for the system, given by

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k) + \mathbf{w}_k \quad (3.18)$$

$$\mathbf{z}_k = h(\mathbf{x}_k) + \mathbf{v}_k \quad (3.19)$$

which represents a minimal change in the KF algorithm, represented in (3.2).

Algorithm 3.2: Extended Kalman Filter

Init: Intialize $\hat{\mathbf{x}}_0, \mathbf{P}_0$
Input: Measurement \mathbf{z}_k
Output: Estimation $\hat{\mathbf{x}}_k$

```

1 for  $k \leftarrow 0$  to  $n$  do
2    $\hat{\mathbf{x}}_k^- = f(\hat{\mathbf{x}}_{k-1}, \mathbf{u}_{k-1})$ 
3    $\mathbf{P}_k^- = \mathbf{A}\mathbf{P}_{k-1}\mathbf{A}^T + \mathbf{Q}$ 
4    $\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}^T (\mathbf{H}\mathbf{P}_k^- \mathbf{H}^T + \mathbf{R})^{-1}$ 
5    $\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k (\mathbf{z}_k - h(\hat{\mathbf{x}}_k^-))$ 
6    $\mathbf{P}_k = \mathbf{P}_k^- - \mathbf{K}_k \mathbf{H} \mathbf{P}_k^-$ 
7 end
```

However, the EKF has one important drawback, i.e. the algorithm can diverge if the initial estimate is wrong, or if the system is incorrectly modeled. This happens because EKF applies a linearization to the model in the form of

$$\mathbf{A} \equiv \left. \frac{df}{dx} \right|_{\hat{\mathbf{x}}_k^-}, \quad (3.20)$$

$$\mathbf{H} \equiv \left. \frac{dh}{dx} \right|_{\hat{\mathbf{x}}_k}. \quad (3.21)$$

In order to do the parameter estimation an augmented model needs to be derived from the final state-space model represented in (3.10). The buck converter load will now change in time, being now called $R_{\text{out}}(t)$. The model for the new state can be considered constant or slowly varying, i.e. $dR_{\text{out}}(t)/dt = 0$. The discrete equations that represent the state transition of the system are given

by

$$i_L(k+1) = i_L(k) - \left(\frac{R_L}{L} + \frac{R_C R_{out}(t)}{L(R_{out}(t) + R_C)} \right) T_s i_L(k) - \frac{R_{out}(t) T_s}{L(R_{out}(t) + R_C)} v_C(k) + \frac{\delta T_s}{L} v_{in}(k) \quad (3.22)$$

$$v_C(k+1) = \frac{R_{out}(t) T_s}{C(R_{out}(t) + R_C)} i_L(k) + i_L(k) - \frac{T_s}{C(R_{out}(t) + R_C)} v_C(k) \quad (3.23)$$

$$R_{out}(k+1) = R_{out}(k). \quad (3.24)$$

EKF alongside the state update equations needs an output update equation, which establishes a relation between the systems states and the measured variables. In this case, the measured variable of the system will be the output voltage $v_{out}(k)$ and the states are $i_L(k)$, $v_C(k)$ and $R_{out}(k)$, meaning that none of the states will be directly measured.

Analyzing the circuit, $v_{out}(k)$ can be written as the sum of the voltage at the terminals of the capacitor C and the voltage drop across R_C . The equation for the system output in terms of the system states is then

$$v_{out}(k) = \frac{R_C R_{out}(k)}{R_{out}(k) + R_C} i_L(k) + v_C(k) - \frac{R_C}{R_{out}(k) + R_C} v_C(k). \quad (3.25)$$

Equations (3.20) and (3.21) show that the state and output matrices are represented by the derivatives of the respective functions, evaluated with the last estimate results. The models for the state transition matrix \mathbf{A} and the output vector \mathbf{H} are finally given by

$$\mathbf{A} = \frac{\partial f}{\partial \mathbf{x}}(\mathbf{x}(k), \mathbf{u}(k)) = \begin{bmatrix} 1 - \frac{R_L T_s}{L} - \frac{R_C T_s R_{out}(k)}{L(R_{out}(k) + R_C)} & -\frac{R_{out}(k) T_s}{L(R_{out}(k) + R_C)} & -\frac{R_C^2 i_L(k) T_s}{(R_{out}(k) + R_C)^2} - \frac{R_C v_C(k) T_s}{(R_{out}(k) + R_C)^2} \\ \frac{R_{out}(k) T_s}{C(R_{out}(k) + R_C)} & 1 - \frac{T_s}{C(R_{out}(k) + R_C)} & \frac{R_C i_L(k) T_s}{C(R_{out}(k) + R_C)^2} \\ 0 & 0 & 1 \end{bmatrix} \quad (3.26)$$

$$\mathbf{H} = \frac{\partial h}{\partial \mathbf{x}}(\mathbf{x}(k)) = \begin{bmatrix} \frac{R_C R_{out}(k)}{R_{out}(k) + R_C} & 1 - \frac{R_C}{(R_{out}(k) + R_C)} & \frac{R_C^2 i_L(k)}{(R_{out}(k) + R_C)^2} + \frac{R_C v_C(k) T_s}{(R_{out}(k) + R_C)^2} \end{bmatrix} \quad (3.27)$$

EKF equations take into consideration the measurement noise, \mathbf{v}_k , along with the process noise, \mathbf{w}_k , which are assumed to be Gaussian white noise, i.e., a zero mean signal with a known covariance, given by

$$\mathbf{v}_k \sim \mathcal{N}(0, \sigma_v^2) \quad (3.28)$$

$$\mathbf{w}_k \sim \mathcal{N}(0, \sigma_w^2). \quad (3.29)$$

The covariance matrices of the noise signals are calculated as

$$\mathbf{R} = E[\mathbf{v}_k \mathbf{v}_k^T] \quad (3.30)$$

$$\mathbf{Q} = E[\mathbf{w}_k \mathbf{w}_k^T] \quad (3.31)$$

where \mathbf{R} represents the measurement noise covariance and \mathbf{Q} the process noise covariance. \mathbf{R} represents the standard deviation of the noise on the system measurements. This means that this matrix is often calculated based on observations or even with dynamic updating. \mathbf{Q} represents the process noise covariance, or in other words, the confidence in the model and it is directly related to \mathbf{R} , e.g., if \mathbf{Q} is bigger than \mathbf{R} it means that the algorithm has more confidence in the measurements than in the model of the system. Considering this, \mathbf{R} will be determined analyzing the noise on v_{out} measured data. \mathbf{Q} estimation is not so trivial but some assumptions can be done looking carefully at the meaning of the parameters. The measured variable is v_{out} and this particular variable is not represented by any of the system states. However, considering the steady state behavior, the current that flows through the capacitor C is almost 0, so a fair approximation can be done considering that $v_C \approx v_{out}$ in steady state. Therefore, it can be assumed that the v_C is in the majority of the time measured and this parameter covariance should be higher than the covariances of i_L and R_{out} , since the algorithm should trust more in the measured value and less in the buck converter model.

Definition of the initialization parameters is the last step to implement the EKF algorithm. The error covariance \mathbf{P} and the initial estimate \mathbf{x}_0 determine how fast the algorithm can converge the correct value, requiring a fine tuning. This initialization parameters can be defined with typical data or with educated guesses. \mathbf{x}_0 is composed by the initial estimation of the parameters. The value of the first measurement from v_{out} can be considered as the initial estimate for the voltage at the capacitor, v_C . The estimation is not very accurate, but this value is a better estimate for the initial value of the voltage than any other. This assumption implies also that the value of the initial covariance for v_C will be lower than the other two parameters. The definition of the remaining parameters will be done using, as stated before, empirical data.

The model from section 3.1 and the EKF algorithm presented in this section concludes the development of the current observer. EKF algorithm developed uses the model information to estimate the inductor current, which will be used in the control algorithm presented in the next section.

3.3 Control Algorithm

Predictive Current Control algorithms have two control loops: the voltage loop (outer loop) and the current loop (inner loop). The voltage loop compares the measured voltage with the reference one, providing then the reference current to the current loop. Usually this loop is a simple proportional-integral (PI) controller, represented by the following equations:

$$I(k) = K_i \cdot e + I(k-1) \quad (3.32)$$

$$U(k) = K_p \cdot e + I(k) \quad (3.33)$$

where e is the voltage error, given by the difference of the measured value and the reference, K_p and K_i are, respectively, the proportional and integral coefficients, $I(k)$ is the integral error and $U(k)$ is the PI controller output.

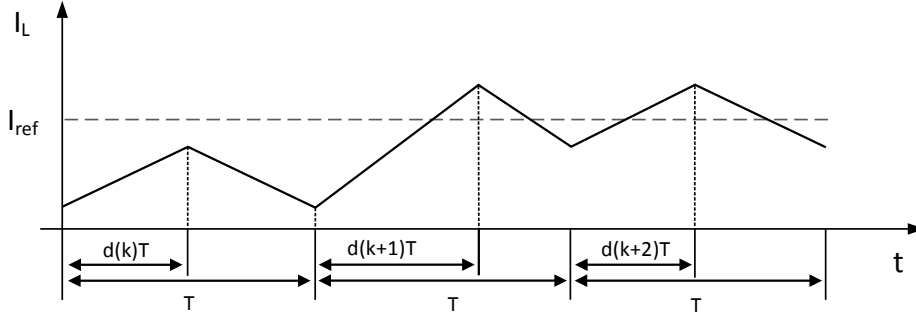


Figure 3.2: Current waveform considering the proposed method.

In this particular implementation of the predictive current control the current loop compares the reference current with the estimated one provided by EKF, using the result of the comparison to adjust the PWM duty cycle, compensating the observed difference.

Average value of the inductor current is the control target, so it is important to analyze the behavior of this variable across the switching cycles. The considered control variable is an averaged value but in fact the current has ripple with known slopes. The slopes of the current can be written by

$$s_1(k) = \frac{v_{in}(k) - v_{out}(k)}{L} \quad (3.34)$$

$$s_2(k) = \frac{v_{out}(k)}{L} \quad (3.35)$$

where $s_1(k)$ denotes the positive slope and $s_2(k)$ the negative slope of the input current.

Section 3.1 emphasizes the importance of an accurate model for the estimation of the buck converter inductor current and resistor. Using the same approach for the current slope modeling, the presented model is simplistic and does not consider the equivalent series resistor of the inductor, which has a strong influence in the current, since all the current that flows through the inductor also flows through the resistor. Equations for the current slopes can then be updated to

$$s_1(k) = \frac{v_{in}(k) - v_{out}(k) - R_L i_L(k)}{L} \quad (3.36)$$

$$s_2(k) = \frac{v_{out}(k) + R_L i_L(k)}{L}. \quad (3.37)$$

The inductor current rises during the on time of the buck converter, i.e., during δ and falls during the off time, or $1 - \delta$. Variations on the current in two consecutive switching cycles can be derived using (3.36) and (3.37), giving

$$i_L(k+1) - i_L(k) = s_1(k) \delta T_s - s_2(k) (1 - \delta) T_s. \quad (3.38)$$

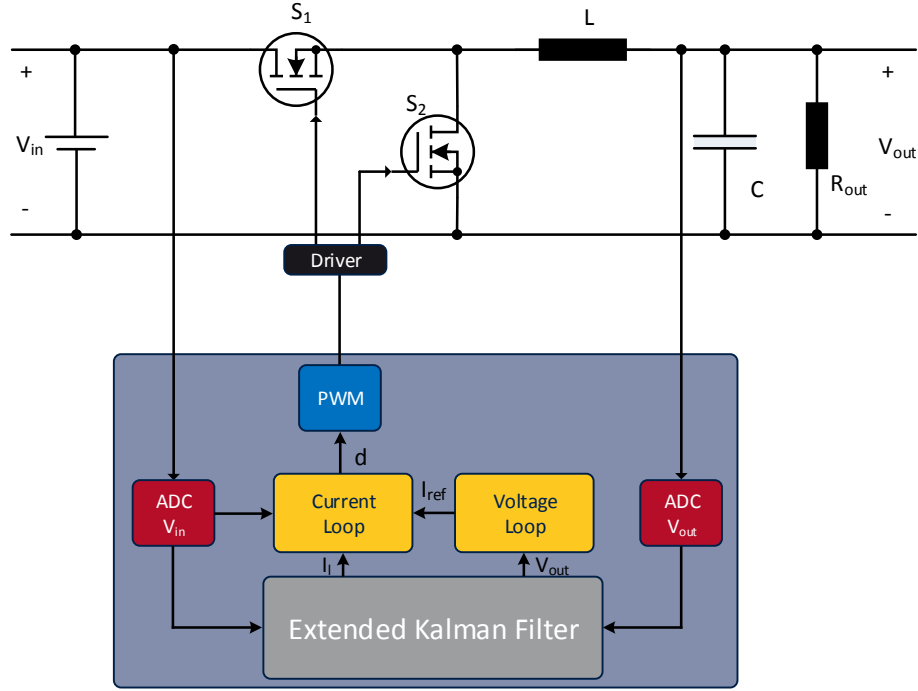


Figure 3.3: Proposed predictive current control loop.

When there is a disturbance on the current, the duty cycle has to be adjusted to compensate that change and reach the reference current I_{REF} . Fig. 3.2 illustrates this behavior.

Solving (3.38) in order to the duty cycle represents the value that has to be given to the PWM in order to reach the desired level of current. Assuming this fact, $i_L(k+1)$ is the next value of the current, i.e., the value that the controller has to achieve – the reference value of the current I_{REF} given by the voltage control loop. The duty cycle that adjusts the current error can be calculated by

$$\delta = \frac{I_{REF} - i_L(k) + s_2(k) T_s}{(s_1 + s_2) T_s}. \quad (3.39)$$

Equation (3.39) represents the control-loop algorithm, which takes into account the measured parameters v_{out} and v_{in} , the reference current provided by the control loop, and the estimated current by the current observer. With this, the full control algorithm for implementing predictive current control is derived and can be represented in Fig.3.3.

One important thing to retain from Fig. 3.3 is that, unlike other predictive control algorithms, the output voltage does not come directly from the ADC. This means that the voltage is first filtered by EKF, which improves the control loop immunity to output voltage noise.

3.4 Summary

In this chapter the algorithm for the predictive current control loop is presented. In the first section the model of the buck converter considering the parasitic parameters is derived, using the technique of SSA. With this model, the average values of the output voltage and inductor current can be represented for the whole switching cycle. In the second chapter an augmented state EKF is developed, allowing both inductor current and load resistance estimation. The last section presents the proposed average current control loop that makes use of the current estimation presented in the first two sections.

Chapter 4

Experimental Results

The present chapter presents the experimental validation of the theoretical derivation of the current-control algorithm introduced in the previous chapter. It starts by describing the testing setup, including hardware and software descriptions and the test procedures. Simulations with real data were performed, allowing simple prototyping and tuning, using MATLAB environment. Such results are presented in the second section of the current chapter. Finally, the last section reports the algorithm implementation on a microcontroller unit, evaluating the estimations accuracy and step response performance.

4.1 Setup Description

The proof-of-concept device is a synchronous buck converter kit, provided by Infineon Technologies (Fig. 4.1).

The kit is composed by a hardware board and a control card, in this case a XMC4200 with an ARM-CORTEX M4 processor running at 80 MHz. The board has two similar buck converters that can operate also in interleaved mode. In the context of this dissertation, only one buck converter will be used to perform the data validation. Design specifications of the buck converter and the main component specifications are presented in the tables 4.1 and 4.2, respectively.

Table 4.1: Buck converter design specifications.

Input Voltage	15 V
Output Voltage	1.8 V
Maximum Output Current	5 A
Switching Frequency	100 kHz

Table 4.2: Buck converter parameter specifications.

Inductor	33 mH
Inductor winding resistance ²	71 m Ω
Capacitor	330 μ F
ESR of the capacitor	100 m Ω

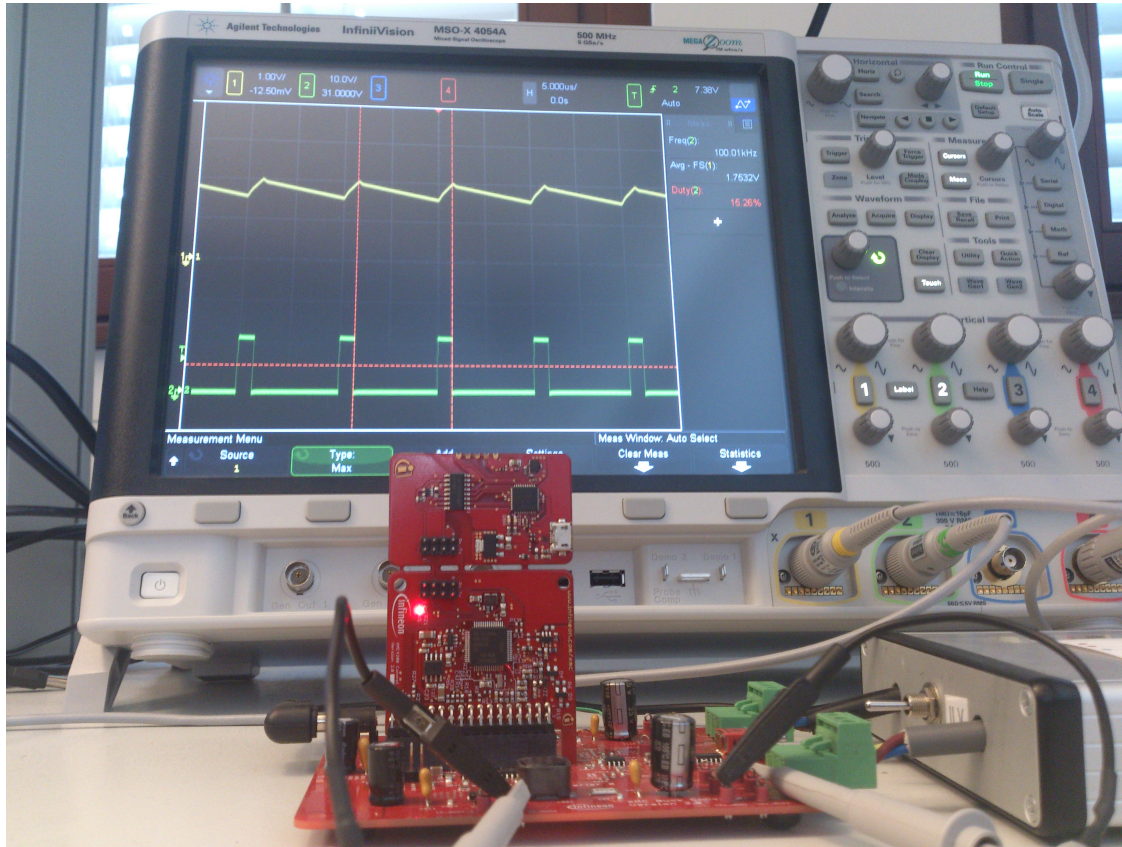


Figure 4.1: Infineon XMC buck converter kit.

The buck converter board has Infineon BSZ060NE2LS OptiMOS power MOSFETs, BOURNS SRF1280-330M inductors and PANASONIC electrolytic capacitors. Alongside these specifications and components, the board also has a step-load switch, which can be used to easily introduce a load-step in the buck converter. Lastly, the inductor current can be sensed using the built-in current sensing circuit, composed by a $15\text{ m}\Omega$ shunt resistor and a LM358D operational amplifier.

XMC4200 has a diverse set of peripherals that can be used to do SMPS control. The ones used in this project are the timers, in specific the capture/compare unit 8 (CCU8) that provides signals with 12.8 ns resolution is used with an high-resolution pulse width modulation (HRPWM) module, generating then PWM waves with a resolution of 150 ps and the versatile analog-to-digital converter (VADC), having a 12 bit resolution. The HRPWM module is used to generate the control signal for the control switch, as well as the inverted signal for the synchronous switch, implementing also a dead time interval to avoid short circuits in the circuit. VADC is used to sample the input and output voltages, having some data accumulation features that will be discussed later.

The platform used to configure the peripherals, write the code and download it to the microcontroller is DAVE4, an Eclipse based integrated development environment (IDE) developed by Infineon and launched as productive version on July 2015. DAVE has a set of APPs that can be

²The shunt resistance presence described means that the buck converter circuit has an additional resistor in series with the inductor, which can be added to the inductor winding resistance on table 4.2. So in practical implementation, the shunt resistor is added to this parameter.

configured in a GUI interface, allowing a faster and intuitive peripheral configuration. The APP used in this dissertation was the buck converter control, released as beta version. This APP allows the configuration of the buck converter switching frequency, control rate of the control loop, ADC channels to be converted, providing also overload protection. The APP diagram is shown in Fig. 4.2, where some peripherals referred above are also represented.

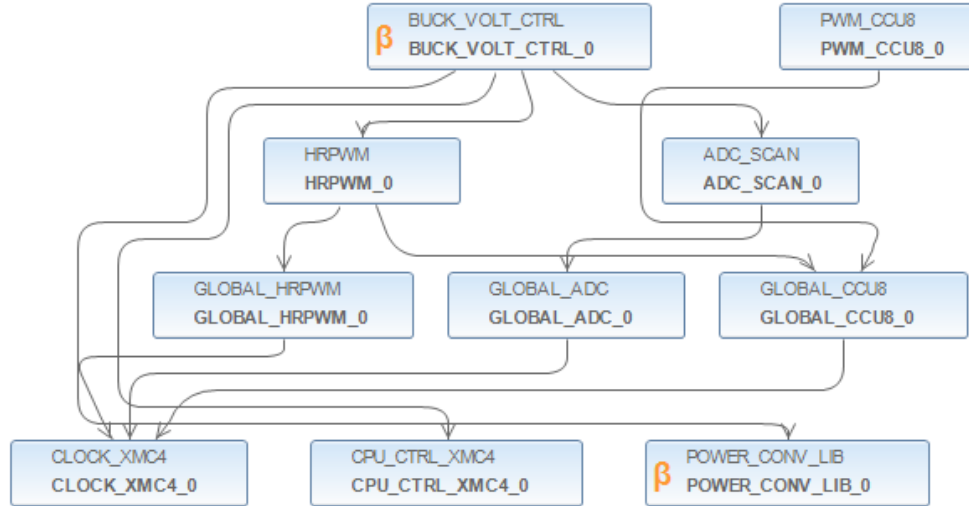


Figure 4.2: Diagram of the peripherals used by the microcontroller unit.

Debugging was done using the Eclipse plugin winIDEA Open, developed by iSYSTEMS, which allows logging of the debug variables in a text file format, which can be easily processed, analyzed and plotted in MATLAB.

The data logging was done using 4 arrays that can store all the important variables for the algorithm: output voltage, input voltage, duty cycle and inductor current. The first three are inputs to the algorithm and the inductor current is logged as a reference for the estimation parameters. Each value is logged once per switching cycle until the 1500th switching cycle is hit – an observed number of cycles that allow the buck converter voltage to achieve steady state. The debugger is then paused and the arrays exported to a text file, which is processed and exported to a MATLAB data file. All the simulations are done in MATLAB, allowing a simple and quick prototyping, tuning and data analysis.

As the considered device is a synchronous buck converter, the control PWM waves for both the control and synchronous switch have to be synchronized and cannot be turned on/off at the same time, due to the short circuit possibility. On the other hand, the dead time should not be too long, because it will lead to unwanted power losses. Modeling the behavior of the buck converter in the dead time is not straightforward because there are many parasitic capacitances and, moreover, the behavior is not the same for all the load resistors. This difficulties led to an offline fixed dead time adjustment, which was performed following these steps:

- The dead time was initialized with 200 ns;
- The dead time value was decreased by a 12.5 ns and the duty cycle value was logged;

- The supramentioned step is repeated while the duty cycle value continues decreasing;
- When the duty cycle value increases, the optimal dead time is achieved.

These steps were performed to both rising and falling dead times, leading to a optimized dead times of 25 ns for both dead times.

4.2 Current Estimation Simulations

During this simulations, the tuning of the EKF was done evaluating the performance results with the parameter variations. This tuning was done following the considerations made in the section 3.2.

The first determined parameter was the measurement noise covariance, \mathbf{R} , which can be extracted from the output voltage calculating the standard deviation of the signal, using for instance the command *std* from MATLAB. This matrix is a single value: 0.602.

To determine the initial estimation vector, \mathbf{x}_0 , the initial current was analyzed in diverse experiments, which lead to very different results. Therefore, the chosen value for the inductor current initial estimation result was 0. For the specified simulations, the tested loads are $0.5\ \Omega$ and $1\ \Omega$, then the initial estimate for the load was the mean value of these – $0.75\ \Omega$. The initial estimate for the capacitor voltage will be the first value of the output voltage, defining then

$$\mathbf{x}_0 = [0, v_{\text{out}}(1), 0.75]^T. \quad (4.1)$$

Is important to note that an accurate initial estimation provides to EKF a faster convergence time in the first instants of the algorithm execution, but is not determinant to the algorithm convergence or stability in the long term.

The remaining parameters to tune are the noise process covariance matrix \mathbf{Q} and the error covariance \mathbf{P} . First the \mathbf{Q} matrix was fixed and \mathbf{P} was adjusted assuming the considerations made in 3.2 and taking into consideration the algorithm results. The final matrices are

$$\mathbf{Q} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1.5 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.2)$$

$$\mathbf{P} = \begin{bmatrix} 10 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 10 \end{bmatrix}. \quad (4.3)$$

After the initial EKF tuning, some simulations with real data were done. The first simulation assumed a constant load of $0.5\ \Omega$ and evaluated the accuracy of the estimation, comparing it to the real data. Fig. 4.3 shows the obtained results, in a time interval of 15 ms.

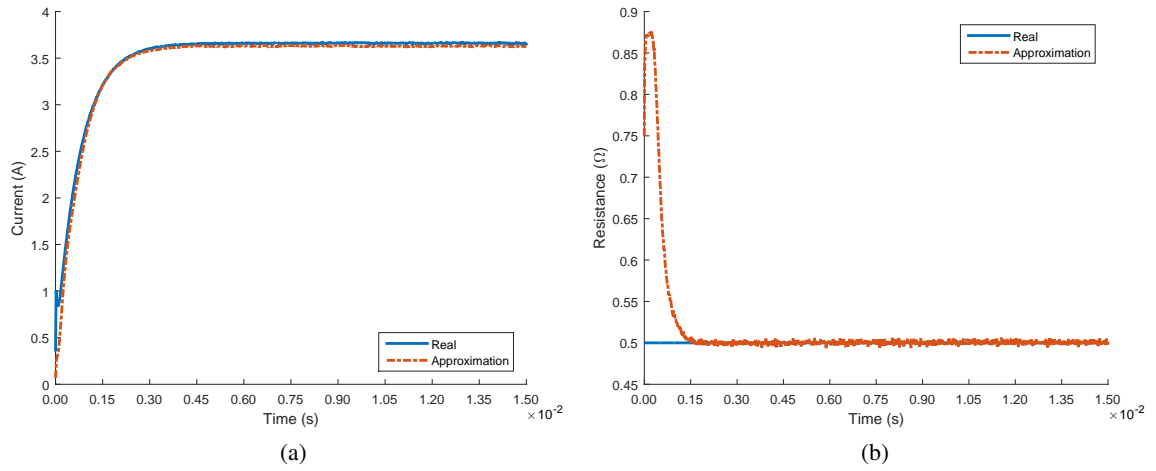


Figure 4.3: EKF estimations and comparison with the real values for a 0.5Ω load: (a) inductor Current (b) load Resistor.

Analyzing the obtained results, the final average value for the estimated current is 3.454 A, while for the real current is 3.495 A, giving a final estimation error of 1.17 %. For the load the estimation error is 2.92 %, which represents a considerably large error in comparison with the current error. However the plotted data shows that in the first EKF iterations the load is unknown and presents a large error comparing to the steady state values. The steady state errors for both estimations are: inductor current – 0.90 %; load resistor – 0.14 %.

EKF is a recursive algorithm and contains an element that acts like a memory, the Kalman gain. This gain is calculated in each iteration of the algorithm and represents the confidence in the predicted value, i.e. how much the measurement will influence the final estimation. Gain values will tend to a constant value, lower than the value at the beginning of the filter initialization, meaning that sudden measurement changes will slowly affect the next estimates. The second simulation was done assuming a step load on the buck converter, in order to test the behavior described above. Results with a step load from 0.5Ω to 1Ω were evaluated and are represented in Fig. 4.4.

Results show that the filter slowly adapts to the load change, taking about 3.4 ms in order to converge to the real value. Standard buck converters take a few hundreds of microseconds to regulate the output voltage, meaning that if the delay of the estimation is in the range of milliseconds, the whole estimation time could be in the range of tens of milliseconds, being unacceptable to most of buck converter control requirements. To overcome this issue a dynamic error covariance update is implemented.

Section 3.2 mentions that the error covariance matrix \mathbf{P} determines the error between the estimated value and the measured value, tending to be lower with the EKF execution. The proposed approach suggests the reset of this variable to the initial value when a certain threshold is crossed. The chosen threshold is the measured output voltage: when this voltage suffers a change of 10 % comparing to the last switching cycle, the error covariance matrix is updated to the initial value,

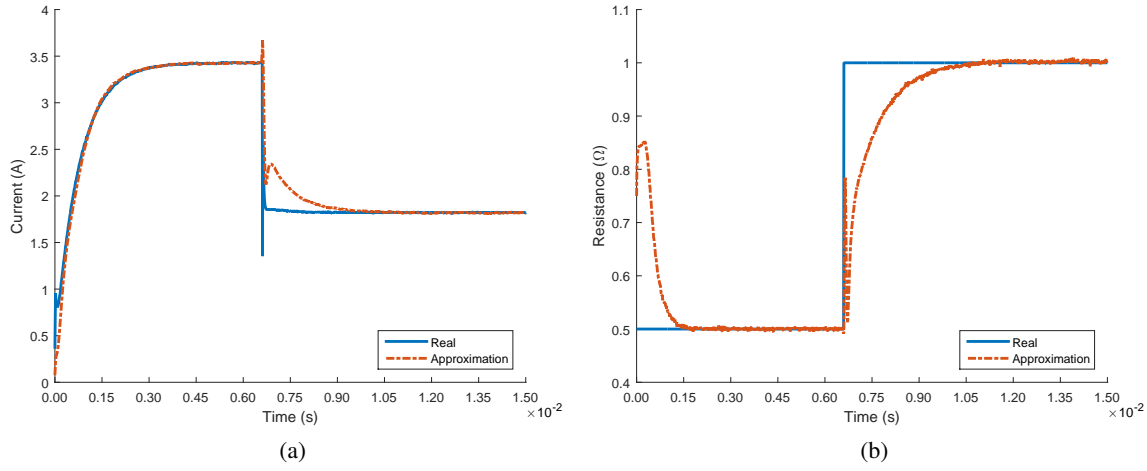


Figure 4.4: EKF estimations and comparison with the real values for a 0.5 Ω to 1 Ω step load: (a) Inductor Current (b) Load Resistor.

already presented in 4.3. Results of the same simulation with the proposed method are presented in Fig. 4.5, achieving improved transient response of the algorithm.

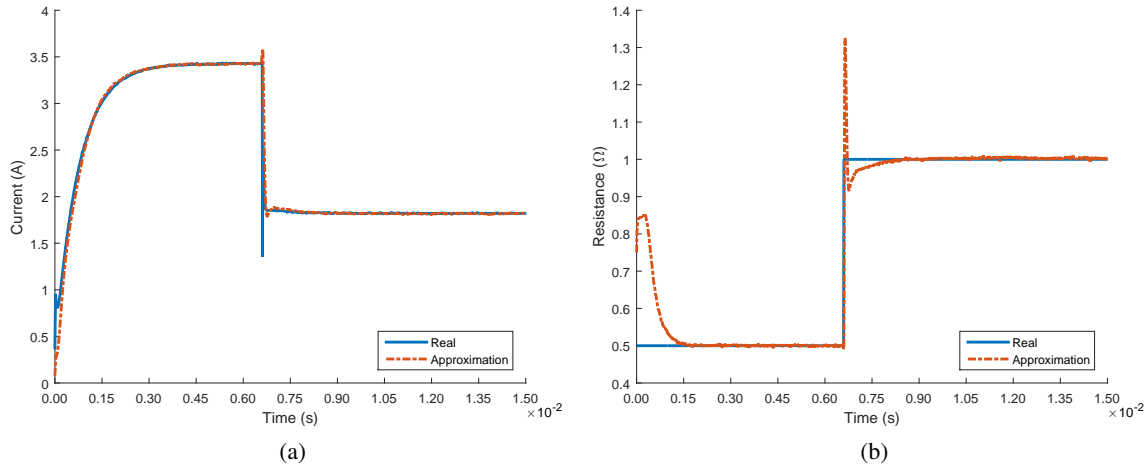


Figure 4.5: EKF estimations and comparison with the real values for a 0.5 Ω to 1 Ω step load with dynamic error covariance update: (a) inductor Current (b) load Resistor.

As stated in Chapter 2, when compared to other common predictive control methods, EKF shows better performance when the output voltage is affected by noise. The voltage that serves as input to the control loop is filtered first, having a minimum influence in the current estimation algorithm. To evaluate the performance improvement achieved by EKF in this situation, noise was artificially injected in the output voltage and the results were evaluated. Two different magnitudes of noise were injected in the output signal: one with a distribution given by $\mathbf{X} \sim \mathcal{N}(0, 0.001)$ and other given by $\mathbf{Y} \sim \mathcal{N}(0, 0.005)$. The results were evaluated and are represented in Fig. 4.6.

Results show that EKF is capable of reducing significantly the impact of the output voltage noise in the control loop response. In Fig. 4.6a the oscillations without EKF can achieve the

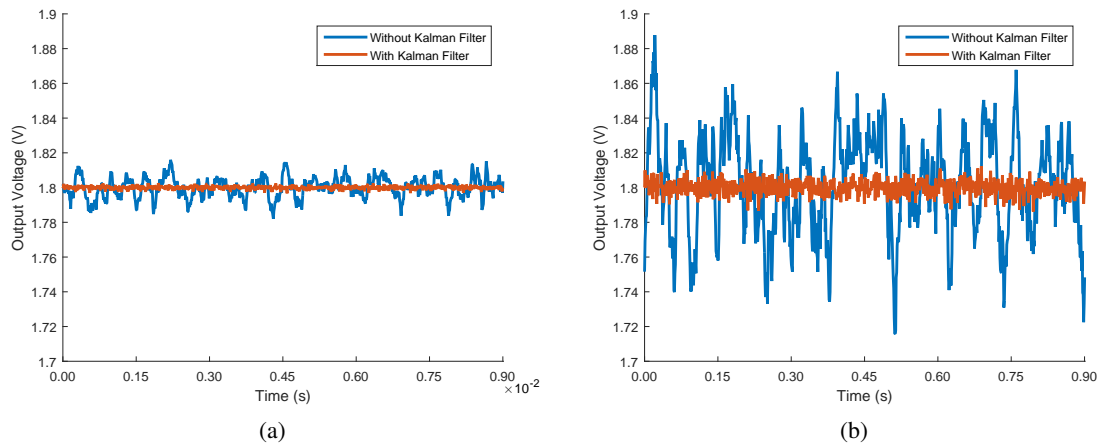


Figure 4.6: Output voltage values when affected by noise: (a) distribution X (b) distribution Y.

15 mV mark, while with EKF the oscillations never go up from 5 mV; in Fig. 4.6b oscillations in the voltage without EKF can achieve almost 90 mV and with EKF the oscillations drop to around 15 mV. This shows the effectiveness of the proposed method when the output voltage is affected by noise, reducing its influence on the control loop.

This simulation concludes this section and in the next section the algorithm will be implemented in the microcontroller unit and the results will be evaluated and compared to the results obtained in this section.

4.3 Implementation of the Algorithm in the Microcontroller

The executed simulations were planned and coded to enable the prototyping and tuning of the algorithm in a simple way, having also an intuitive way to evaluate the results. Implementing this algorithm on the microcontroller, however, has its own set of challenges, not considered at simulation time.

Microcontrollers do not have by default a way to do matrix operations that are required to the EKF execution. The first thought was to implement a generic library that could be used to do this operations, such as add, multiply or invert matrices. EKF algorithm was implemented in the microcontroller using the created library and performance tests revealed that for this algorithm the execution time was near to 200 μ s, meaning that the algorithm would run every 20th cycle, which is an unacceptable control period leading to very slow response times to load changes. To overcome this issue an optimized version of EKF is developed – a single function that treats all the matrix operations as simple arithmetic operations, considering the individual elements of each matrix. Optimizing the EKF algorithm lead to a 14 μ s execution time of the algorithm.

Another problem that was not considered until now is the measurement of the output voltage. Ideally this voltage is measured in a point that represents the average value of the wave, but this calibration is hard to do and the mentioned point is variable with the load. In this implementation

the feature of data accumulation of the VADC is used, allowing the accumulation of 2 conversion results in the same result register (the module allows up to 4 conversions to be stored). Using this feature is possible get a more precise value for the output voltage, without the need for a software low-pass filter (like a moving average filter).

After implementing these changes and the full algorithm in the microcontroller, experiments with the real setup were performed. The voltage and current loop, as well as reading conversion values and updating the duty cycle take $8\ \mu\text{s}$, which added to the EKF execution time ($14\ \mu\text{s}$) make a total of ($22\ \mu\text{s}$), meaning that the whole control algorithm will run every third switching cycle.

In the first experiment a sweep load was applied to evaluate the accuracy of the estimation in steady state. The applied sweep was from $0.5\ \Omega$ to $2\ \Omega$ with a step of $0.1\ \Omega$ and the results are depicted in Fig. 4.7. As can be observed, both current and load estimations are below 1 %, proving the accuracy of the implemented algorithm for different loads.

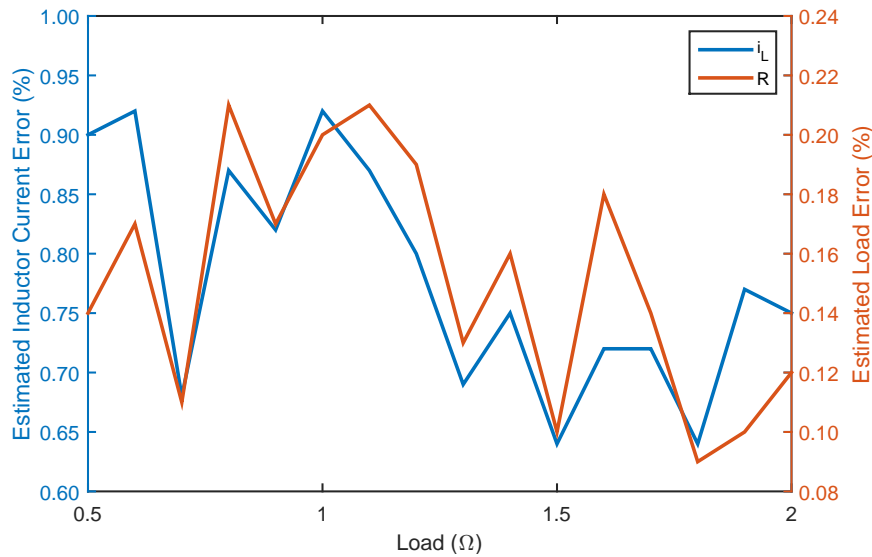


Figure 4.7: Accuracy of the inductor current and load resistor estimations with a sweep load.

The second performed experiment was done with a step load from $0.5\ \Omega$ to $1\ \Omega$ and the results were logged and plotted using MATLAB (Fig. 4.8).

Comparing the measured results with the simulated ones on 4.5 it can be seen that the response times are bigger than the simulation results. These results are understandable due to the fact that the control loop in the microcontroller only runs in a interval of 3 switching cycles, delaying then the estimation. Nevertheless the measured step response shows a similar behavior when compared to the simulated one, meaning that in the normal buck converter operation the delay of the estimation will be lower than the delay on the beginning.

To evaluate the overall performance of the buck converter a set of experiments were performed to evaluate the response of the output voltage and the inductor current. Fig. 4.9 represents output voltage and inductor current response in the test setup described above, which is, a step load from $0.5\ \Omega$ to $1\ \Omega$ and from $1\ \Omega$ to $0.5\ \Omega$.

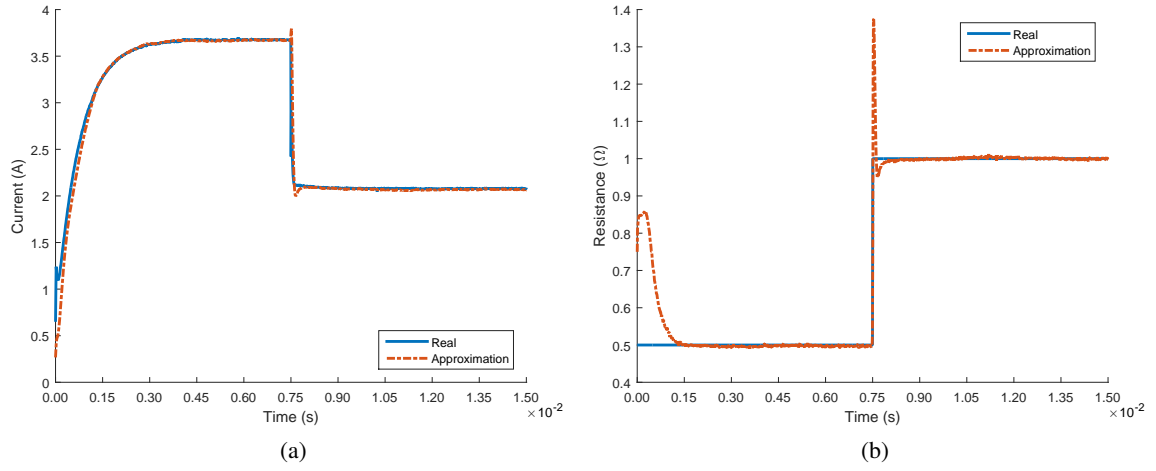


Figure 4.8: Voltage and current step response to a step load from: (a) 0.5Ω to 1Ω (b) 1Ω to 0.5Ω .

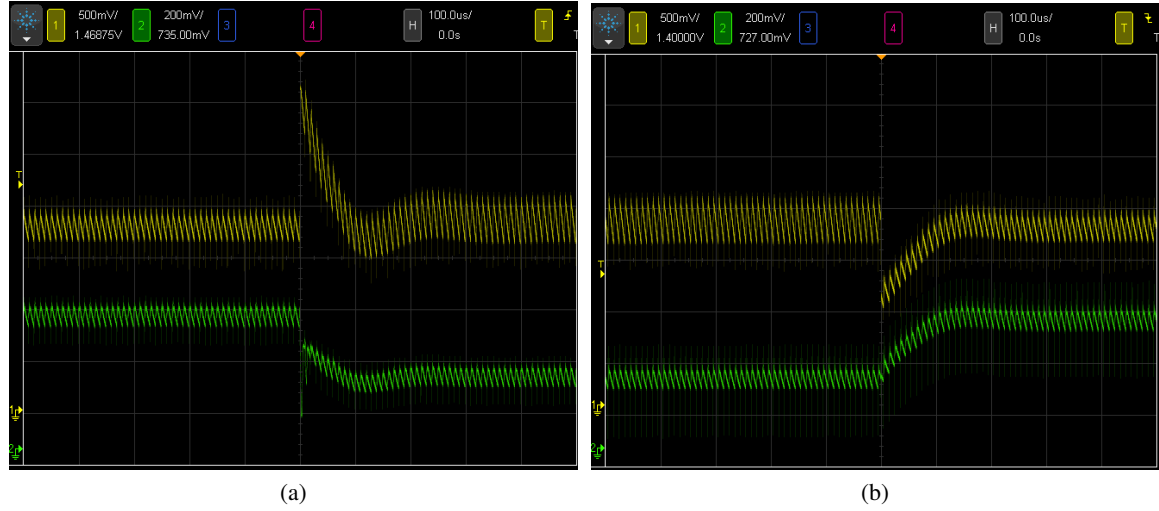


Figure 4.9: EKF estimations and comparison with the real values for a step load in the MCU: (a) inductor Current (b) load Resistor.

Results show that the voltage takes about $200 \mu\text{s}$ to stabilize, showing the ability of the algorithm to correct the load change and maintain a constant voltage of 1.8 V .

The next experimental test was done to evaluate the line regulation capabilities of the proposed algorithm. To perform this evaluation the buck converter board was supplied with a DC power supply, where the input voltage was intentionally changed to evaluate the algorithm response. As stated in section 4.1, the input voltage of this synchronous buck converter is usually 12 V , but in this test the voltage at some point of the simulation intentionally dropped down to 10 V and the results are depicted in Fig. 4.10.

The results confirm that the algorithm is able to adapt to input voltage changing, showing a response time of about $100 \mu\text{s}$, nearly half of the time that the algorithm takes to respond to load changes. This behavior can be explained due to different types of variables that are in considera-

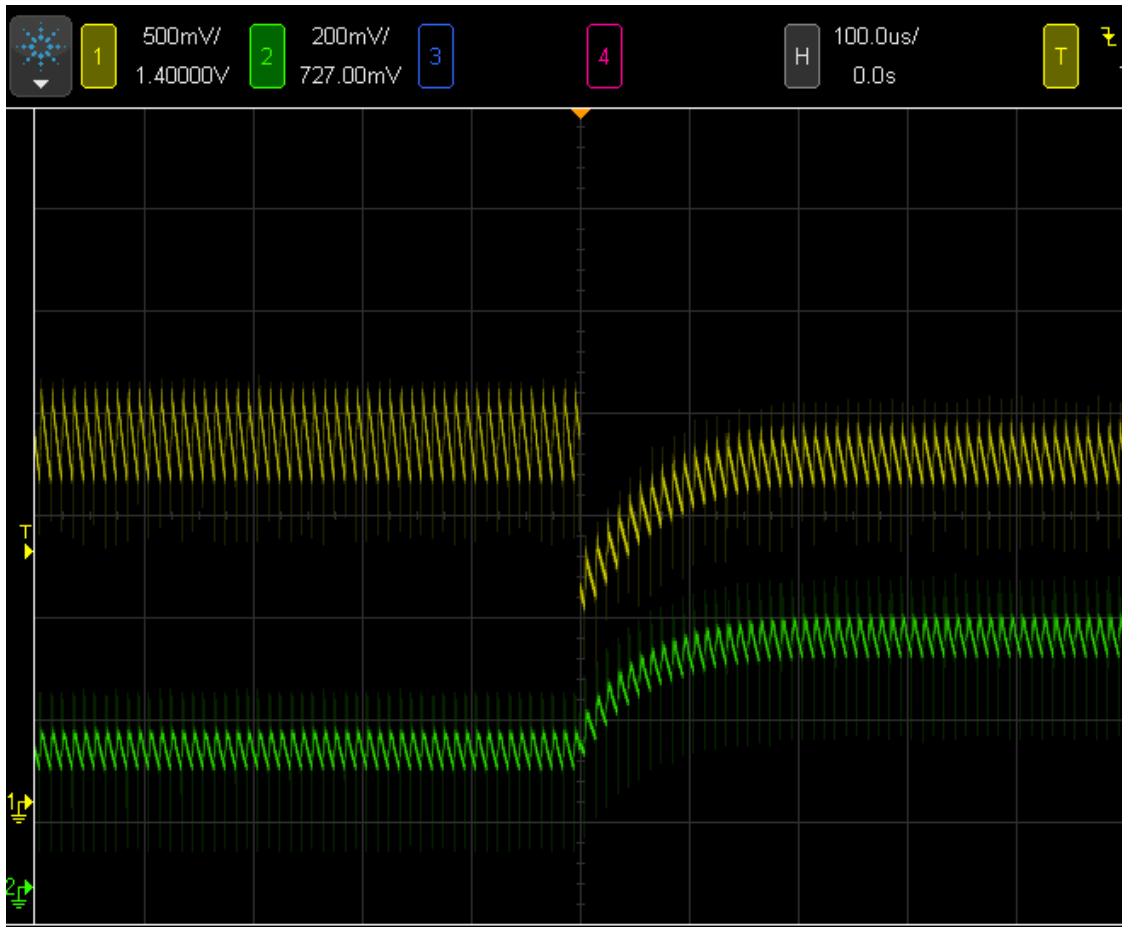


Figure 4.10: Voltage response to a sudden input voltage drop to 10 volt.

tion, that is, a change in the load resistor, R_{out} , and a change in the input voltage, V_{in} . If the state space model derived in Chapter 3 is analyzed, R_{out} is a state and V_{in} is an input. The states are affected by the Kalman gain and as it was mentioned previously this element acts like a memory, showing 'inertia' when the state value quickly changes. The input voltage is not affected by any gain and does not enter in any EKF variables calculation – V_{in} is just part of the system model and any change will affect quickly the response of the algorithm.

Chapter 5

Conclusions and Future Work

In this chapter a balance of the work performed in this dissertation is done, comparing this to the state of the art approaches. Future work suggestions are presented at the end.

5.1 Fulfillment of the Objectives

The objective of this work was improving the performance of a synchronous buck converter using a KF. The nonlinear version of KF, EKF, can be used to design a predictive current control loop that do not need current sensors, reducing also the impact of the noise in the control loop.

First of all, an overview of buck converter control methods was done. Voltage mode control was the first used method due to its simplicity but slow transient response and poor line regulation lead to the appearance of the current mode control. CMC and its variants (peak, valley, average, etc.) have their own pros and cons, but they all require current measurement that is not always straightforward. Predictive current control appeared then as a sensorless (no need to measure the inductor current) method, to overcome the difficulties in measuring the current. A literature review containing the predictive current control methods was performed, which served as base to all the work.

Then the theoretical derivation of the whole predictive current control algorithm based on EKF was done, consisting in 3 main parts: deriving the buck converter model, tune the EKF and design the control loop. In order to derive the model a method of state-space averaging was employed, which allows a division of the buck converter operation in two states that are merged to provide the system model. The system also has in consideration some parasitic parameters in order to improve the accuracy of the model. The EKF equations and parameters are defined after the model is derived and the combination between them composes the current observer, which will be responsible to estimate the inductor current, as well as the output voltage. In the last part the control loop is designed, based on the equations of the inductor current slopes.

Next an experimental setup was assembled to test the predictive current control algorithm. Simulations and experiments show an overall estimation error of the current and resistor of approximately 1 %, while considering the steady state, this error drops to less than 1 %. The output

voltage contains noise, which may affect the response of the control loop, causing undesired oscillations around the mean value of the output voltage. The EKF approach shows a better performance when the output voltage contains noise, adding to the fact that it can also provide a fast line regulation.

Overall, the predictive current control method can be compared to the previous state of the art methods having in consideration:

- Step load response time: the proposed algorithm transient response after a load change is about 200 μ s, showing a similar response time to the methods presented in [21, 5], showing a better step response than the regular voltage mode control method using a PI in the control loop;
- Provides line regulation to input voltages changes: all the reviewed methods provided line regulation, which was intrinsic to the method or added as a feedforward mechanism. Comparing to the standard control methods, line regulation is achieved by peak current control and average current control, while voltage mode control and valley current control require additional methods to perform this;
- The estimation error considering all the operation time is about 1 % and in steady state an error less than 1 %. The error in the estimation achieved in [5] is of 5 % in steady state, showing an improvement of the accuracy of the algorithm when the EKF method is used;
- The algorithm improves the control loop response in the presence of noise in the output voltage. This improvement is also achieved in [22] and in the second method presented by [20], because they also apply the EKF approach. This noise influence reduction is mainly observed when comparing with other predictive control methods that don't use EKF, since they use equations to estimate the variables.

Overall, despite of being more expensive computationally than other predictive methods, the proposed approach can provide better estimation results, while still maintaining a good transient response. Comparing to all available methods, the EKF approach is the only one that can filter out noise present in the output voltage due to its intrinsic behavior, not needing external filters to do that. Analyzing the results, it can be concluded that the performance of the synchronous buck converter was improved, especially when the output voltage is affected with noise.

5.2 Future Work

This work achieves the proposed goals, but can be improved in some ways. In this section, suggestions for future work will be suggested.

- The optimized dead time can be calculated during runtime. One possibility is to implement a dynamic sensorless method that calculates the dead time. The other possibility is to model

the converter behavior during the dead time period, achieving a third state that can then be joined with the two proposed ones, using state-space averaging.

- More parasitic parameters of the synchronous buck converter can be modeled to improve the estimation accuracy, such as the switch resistor during on time, or the inductance of the capacitance. However the addition of this variables can mean that more states can be added to the system model, increasing the computational requirements.
- The measurement noise covariance can be updated during runtime. Buck converters integrated in a power system can be affected by different magnitudes of noise during the time of operation. If these have considerable changes during the operation time, a dynamic update can be consider to improve the performance of the method.
- Extend the approach to other converter topologies. A synchronous buck converter was the studied type of SMPS but with a modified model the method can be applied to other types of converters, such as boost or buck-boost.

Bibliography

- [1] H. J. Zhang, “Basic concepts of linear regulator and switching mode power supplies,” Linear Technology, Application Note 140, October 2013.
- [2] R. Nowakowski and N. Tang, “Efficiency of synchronous versus nonsynchronous buck converters,” *Analog Applications Journal*, pp. 1–6, 2009.
- [3] “Input and output noise in buck converters explained,” Maxim Integrated, APP 986, Jul 2002.
- [4] R. M. Kotecha, “Analysis and comparison of popular models for current-mode control of switch mode power supplies,” MSc Thesis, Wright State University, 2010.
- [5] M. Run, C. Chen, Z. X. Dong, and Z. X. Cheng, “Optimal current observer for predictive current controlled buck dc-dc converter,” in *International Symposium on Computer, Consumer and Control (IS3C’2014)*, Jun 2014, pp. 896–899.
- [6] J. W. Kimball, “Digital control techniques for switching power converters,” Ph.D. dissertation, University of Illinois at Urbana-Champaign, 2007.
- [7] R. Kalman, “A new approach to linear filtering and prediction problems,” *Transactions of the ASME-Journal of Basic Engineering*, vol. 82, no. Series D, pp. 35–45, 1960.
- [8] Y.-F. Liu, E. Meyer, and X. Liu, “Recent developments in digital control strategies for DC/DC switching power converters,” *IEEE Transactions on Power Electronics*, vol. 24, no. 11, pp. 2567–2577, Nov 2009.
- [9] R. Dhanasekaran, S.; Kumar, E.; Vijaybalaji, “Different methods of control mode in switch mode power supply – A comparison,” *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, vol. 3, no. 1, pp. 6717–6724, 2014.
- [10] R. Mammano, “Switching power supply topology voltage mode vs. current mode,” Unitrode, Design Note 62, 1994.
- [11] M. Zimnik, “Comparison of PWM voltage and current mode control schemes vs. improved hysteretic mode control in switch mode power supplies (SMPS),” Texas Instruments, Inc., White paper.

- [12] R. Sheehan, "A new way to model current-mode control – Part one," *Power Electronics Technology*, pp. 14–20, May 2007.
- [13] D. Meeks, "Loop stability analysis of voltage mode buck regulator with different output capacitor types – Continuous and discontinuous modes," Texas Instruments, Application Report SLVA301, Apr 2008.
- [14] R. Ridley, "Designer's Series: Current-mode control modeling – Part V," *Switching Power Magazine*, pp. 1–12, 2006.
- [15] L. Dixon, "Average current mode control of switching power supplies," Unitrode, Application Note U-140, pp. 356–369.
- [16] R. Sampath, "Digital peak current mode control of buck converter using MC56F8257 DSC," Freescale Semiconductor, Application Note 4716 Rev. 1, May 2013.
- [17] G. Hariman and C. Richardson, "Control method solves low duty-cycle dilemmas," *Power Electronics Technology*, pp. 22–30, Sep 2006.
- [18] P. Midya, P. Krein, and M. Greuel, "Sensorless current mode control-an observer-based technique for dc-dc converters," *Power Electronics, IEEE Transactions on*, vol. 16, no. 4, pp. 522–526, Jul 2001.
- [19] P. Mattavelli, "Digital control of DC-DC boost converters with inductor current estimation," in *IEEE Nineteenth Annual Applied Power Electronics Conference and Exposition (APEC'2004)*, vol. 1, 2004, pp. 74–80.
- [20] A. Beccuti, S. Mariethoz, S. Cliquennois, S. Wang, and M. Morari, "Explicit model predictive control of dc-dc switched-mode power supplies with extended kalman filtering," *IEEE Transactions on Industrial Electronics*, vol. 56, no. 6, pp. 1864–1874, June 2009.
- [21] Y. Qiu, H. Liu, and X. Chen, "Digital average current-mode control of PWM DC-DC converters without current sensors," *IEEE Transactions on Industrial Electronics*, vol. 57, no. 5, pp. 1670–1677, May 2010.
- [22] Q. Zhang, Q. Tong, and H. Zhang, "An inductor current observer based on improved EKF for DC/DC converter," in *International Symposium on Computer, Consumer and Control (IS3C'2014)*, Jun 2014, pp. 892–895.
- [23] B. Johansson, "Improved models for DC-DC converters," Licenciate Thesis, Lund University, 2003.
- [24] R. D. Middlebrook and S. Ćuk, "A general unified approach to modelling switching-converter power stages," in *Proceedings of the IEEE Power Electronics Specialists Conference (PESC'1976)*, 1976.

- [25] S. M. Ćuk, “Modelling, Analysis, and Design of Switching Converters,” Ph.D. dissertation, California Institute of Technology, Nov 1977.
- [26] M. R. Modabbernia, F. Kohani, R. Fouladi, and S. S. Nejati, “The state space average model of buck-boost switching regulator including all of the system uncertainties,” *International Journal on Computer Science and Engineering*, vol. 5, no. 2, pp. 120–132, 2013.